



TUGAS AKHIR - TE 145561

***MONITORING PENGELOMPOKAN BARANG PADA LIFT
KONVEYOR MENGGUNAKAN ARDUINO ETHERNET
SHIELD DENGAN MEDIA WIFI***

Erryka Ayu Diah Charisma
NRP. 10311500010016

Dosen Pembimbing
Ir. Josaphat Pramudijanto, M.Eng.

Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT - TE 145561

***MONITORING THE CLASSIFICATION OBJECT ON
CONVEYOR ELEVATOR USING ARDUINO ETHERNET
SHIELD WITH WIFI***

Erryka Ayu Diah Charisma
NOR 10311500010016

Supervisor
Ir. Josaphat Pramudijanto, M.Eng.

*Electrical and Automation Engineering Department
Vocational Faculty
Sepuluh Nopember Institute of Technology
Surabaya 2018*

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul ***“Monitoring Pengelompokan Barang Pada Lift Konveyor Menggunakan Arduino Ethernet Shield Dengan Media Wifi”*** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2018



Erryka Ayu Diah Charisma
10311500010016

-----Halaman ini sengaja dikosongkan-----

**MONITORING PENGELOMPOKKAN BARANG PADA
LIFT KONVEYOR MENGGUNAKAN ETHERNET SHIELD
DENGAN MEDIA WIFI**

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Memperoleh Gelar Ahli Madya Teknik
Pada
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing,



Ir. Josaphat Pramudijanto, M.Eng.

NIP. 19621005 199003 1 003

**SURABAYA
JULI, 2018**

-----Halaman ini sengaja dikosongkan-----

MONITORING PENGELOMPOKKAN BARANG PADA LIFT KONVEYOR MENGGUNAKAN ARDUINO *ETHERNET SHIELD* DENGAN MEDIA *WIFI*

Nama Mahasiswa : Erryka Ayu Diah Charisma
NRP : 10311500010016
Dosen Pembimbing : Ir. Josaphat Pramudijanto, M.Eng.
NIP : 19621005 199003 1 003

ABSTRAK

Pengangkutan barang baik berupa bahan baku ataupun barang jadi saat ini menjadi yang sangat penting dalam sebuah sistem industri. Karena proses pengelompokkan jumlah barang dan memasukkan data yang dilakukan secara manual sering menimbulkan kesalahan perhitungan yang disebabkan oleh *human error*. Maka dari itu, dibutuhkan alat berupa konveyor dengan penambahan *lift* pada konveyor. Serta menggunakan komputer untuk menampilkan hasil dari pengelompokkan jumlah barang.

Para pekerja hanya meletakkan barang di konveyor, kemudian barang melewati salah satu sensor aktif yaitu *scanner barcode* atau sensor *infrared* dan di-*scan*. Sensor memberikan sinyal kepada Arduino Mega2560 untuk memberikan perintah kepada konveyor agar mendistribusikan barang ke tempat yang ditentukan melalui *lift* atau tidak. Untuk proses monitoringnya, menggunakan program pada LabVIEW yang digunakan untuk membaca data yang dikirimkan oleh *ethernet shield* melalui *wifi*. Ketika data telah didapatkan akan diolah menjadi sekumpulan data yang siap untuk ditampilkan.

Setelah dilakukan pengujian, masih terdapat kesalahan berupa ketidaksesuaian barang yang akan dikelompokkan pada konveyor dengan hasil *monitoring* pada LabVIEW. Ketidaksesuaian dalam hal barang sesuai ketinggian dan *barcode* yang di deteksi dengan hasil *monitoring* di LabVIEW. Pengelompokkan barang dapat di *monitoring* dengan jarak hingga 130 meter tanpa adanya halangan. Jika dengan halangan dapat di *monitoring* dengan jarak hingga 15 meter.

Kata Kunci : *Monitoring* Barang, *Lift* Konveyor, *Ethernet Shield*.

-----Halaman ini sengaja dikosongkan-----

MONITORING THE CLASSIFICATION OBJECT ON CONVEYOR ELEVATOR USING ARDUINO ETHERNET SHIELD WITH WIFI

Name of Student : Erryka Ayu Diah Charisma
Number of Registration : 10311500010016
Supervisor : Ir. Josaphat Pramudijanto, M.Eng.
ID Number : 19621005 199003 1 003

ABSTRACT

The transport of object either in the form of raw materials or finished object is now becoming very important in an industrial system. Because the process of grouping the number of object and entering the data done manually often lead to calculation errors caused by human error. Therefore, a conveyor tool is required with the addition of an elevator on the conveyor. And use the computer to display the results of the grouping of object.

The workers just put the object on the conveyor, then the object through one of the active sensor is a scanner barcode or infrared sensor and the scan. The sensor gives a signal to the Arduino Mega2560 to give an order to the conveyor in order to distribute the object to a designated place via an elevator or not. For the monitoring process, use the program in LabVIEW which is used to read data sent by ethernet shield via wifi. When the data has been obtained will be processed into a set of data ready to be displayed.

After testing, there are still errors in the form of non-conformity of items that will be grouped on the conveyor with the results of monitoring at LabVIEW. Non-conformity in terms of goods according to height and barcodes that are detected with the results of monitoring at LabVIEW. Grouping of items can be monitored with a distance of up to 130 meters without any obstacles. If the obstacle can be monitored with a distance of up to 15 meters.

Key Words : Object Monitoring, Elevator Conveyor, Ethernet Shield

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada junjungan kita Nabi Besar Muhammad SAW.

Tugas Akhir ini disusun untuk memenuhi sebagai persyaratan guna menyelesaikan pendidikan Diploma pada Bidang Studi Elektro Industri, Program Studi D3 Teknik Elektro, Jurusan Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul " *Monitoring Pengelompokkan Barang Pada Lift Konveyor Menggunakan Arduino Ethernet Shield Dengan Media Wifi* "

Dengan terselesaikannya Tugas Akhir ini, Penulis menyampaikan terima kasih yang sebesar - besarnya kepada :

1. Kedua orang tua yang senantiasa mendoakan dan memberikan dukungan dengan tulus tiada henti.
2. Kepala Departemen Teknik Elektro Otomasi
3. Ir. Joshapat Pramudijanto, M.Eng. selaku dosen pembimbing.
4. Pak Ghofur selaku pembimbing OJT di Campina Industry.
5. Staff/Karyawan/Dosen Departemen Teknik Elektro Otomasi.
6. Teman - teman D3 Teknik Elektro Hydra yang selalu memberikan doa, bantuan, semangat, dan dukungannya.
7. Semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat diterapkan pada bidang industri serta bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, Juli 2018

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL.....	i
HALAMAN JUDUL.....	iii
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
HALAMAN PENGESAHAN.....	vii
ABSTRAK.....	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan.....	1
1.3 Batasan Masalah.....	2
1.4 Tujuan	2
1.5 Sistematika Laporan	2
1.6 Relevansi	3
BAB II TEORI DASAR.....	5
2.1 Arduino Mega2560	5
2.2 <i>Ethernet Shield</i>	6
2.3 Sistem <i>Monitoring</i>	6
2.4 Arduino IDE.....	6
2.5 <i>Software LabVIEW</i>	8
2.6 Konveyor.....	8
2.6.1 <i>Belt Conveyor</i>	8
2.6.2 <i>Vertical Conveyor</i>	9
2.6.3 <i>Plastic Flain Chain Conveyor</i>	9
2.6.4 <i>Vibrating Conveyor</i>	10
2.6.5 <i>Hanging atau Heat Conveyor</i>	10
2.7 Motor DC	11
2.8 <i>Relay</i>	11

BAB III PERANCANGAN SISTEM	13
3.1 Diagram Fungsional Alat	13
3.2 Perancangan Perangkat Keras (<i>Hardware</i>)	14
3.2.1 Perancangan Konveyor	14
3.2.2 Perancangan Sistem <i>Monitoring</i>	15
3.3 Perancangan Elektronik.....	16
3.3.1 <i>Wiring</i> Modul <i>Relay</i> dan Arduino Mega2560	16
3.3.2 <i>Wiring</i> Modul <i>Relay</i> dan <i>Power Supply</i>	17
3.3.3 <i>Wiring</i> Modul <i>Relay</i> dan Motor DC	19
3.3.4 <i>Wiring Power Supply</i> dan Motor DC	21
3.3.5 <i>Rangkaian Voltage Regulator</i>	22
3.3.6 Perancangan <i>Ethernet Shield</i>	22
3.3.7 <i>Wiring Ethernet Shield</i> , Sensor, dan <i>Wifi</i>	23
3.4 Perancangan Lunak (<i>Software</i>).....	24
3.4.1 <i>Flowchart</i>	24
3.4.2 Penyusunan <i>Ethernet Shield</i> untuk Arduino	27
3.5 Langkah-Langkah Pemograman LabVIEW	29
3.6 <i>Monitoring Sistem Interface</i>	32
3.7 Program Pengelompokkan dan Penjumlahan	34
3.8 Program Animasi Konveyor.....	35
 BAB IV PENGUKURAN DAN ANALISA	37
4.1 Pengujian Koneksi <i>Wifi</i> dengan <i>PC</i>	37
4.2 Pengujian Koneksi Jarak <i>Monitoring</i>	40
4.2.1 Pengujian Koneksi Tanpa Penghalang	40
4.2.2 Pengujian Koneksi Dengan Penghalang	44
4.3 Pengujian Keseluruhan Sistem Pada <i>Software</i> LabVIEW.....	47
4.3.1 Pengujian Sensor <i>Infrared</i>	47
4.3.2 Pengujian <i>Scanner Barcode</i>	57
 BAB V PENUTUP.....	71
5.1 Kesimpulan	71
5.1 Saran	71
 DAFTAR PUSTAKA	73
LAMPIRAN A	A-1
LAMPIRAN B	B-1
LAMPIRAN C	C-1
DAFTAR RIWAYAT HIDUP	D-1

DAFTAR GAMBAR

	HALAMAN
Gambar 2.1 <i>Arduino Mega2560</i>	5
Gambar 2.2 <i>Ethernet Shield</i>	6
Gambar 2.3 <i>Software Arduino IDE</i>	7
Gambar 2.4 <i>Blok Diagram Software LabVIEW</i>	8
Gambar 2.5 <i>Belt Conveyor</i>	9
Gambar 2.6 <i>Vertical Conveyor</i>	9
Gambar 2.7 <i>Plastic Flain Chain Conveyor</i>	10
Gambar 2.8 <i>Horizontal Vibrating Conveyor</i>	10
Gambar 2.9 <i>Vertical Vibrating Conveyor</i>	10
Gambar 2.10 <i>Hanging atau Overhead Conveyor</i>	11
Gambar 2.11 <i>Motor DC</i>	11
Gambar 2.12 <i>Relay</i>	12
Gambar 2.13 <i>Modul Relay</i>	12
Gambar 3.1 <i>Diagram Fungsional Alat</i>	13
Gambar 3.2 <i>Rancangan Konveyor Serta Lift</i>	15
Gambar 3.3 <i>Diagram Fungsional Sistem Monitoring</i>	16
Gambar 3.4 <i>Wiring Modul Relay dan Arduino Mega2560</i>	17
Gambar 3.5 <i>Wiring Modul Relay dan Power Supply</i>	18
Gambar 3.6 <i>Wiring Modu Relay dan Motor DC</i>	19
Gambar 3.7 <i>Rangkaian Motor Penggerak Lift</i>	20
Gambar 3.8 <i>Rangkaian Motor Penggerak Pusher</i>	20
Gambar 3.9 <i>Wiring Power Supply dan Motor DC</i>	21
Gambar 3.10 <i>Rangkaian Voltage Regulator</i>	22
Gambar 3.11 <i>Rancangan Ethernet Shield</i>	22
Gambar 3.12 <i>Wiring Ethernet Shield, Sensor dan Wifi</i>	23
Gambar 3.13 <i>Flowchart Sistem Kerja Pengelompokkan dan Monitoring</i>	26
Gambar 3.14 <i>Alamat IP Address Ethernet Shield</i>	27
Gambar 3.15 <i>Inisialisasi Sensor Infrared</i>	27
Gambar 3.16 <i>Inisialisasi Scanner Barcode</i>	28
Gambar 3.17 <i>Inisialisasi Relay</i>	28
Gambar 3.18 <i>Pengiriman Data Ke Ethernet Shield</i>	29
Gambar 3.19 <i>Software LabVIEW</i>	30
Gambar 3.20 <i>Membuka Lembar Kerja Baru</i>	30
Gambar 3.21 <i>Blok Diagram dan Front Panel</i>	31

Gambar 3.22	Tampilan pada LabVIEW	31
Gambar 3.23	Pemrograman pada LabVIEW	32
Gambar 3.24	Blog Diagram Komunikasi Arduino	32
Gambar 3.25	<i>TCP Open Conection</i>	33
Gambar 3.26	<i>TCP Write</i>	33
Gambar 3.27	<i>TCP Read</i>	33
Gambar 3.28	<i>TCP Close</i>	34
Gambar 3.29	Program Pengelompokkan dan Penjumlahan	34
Gambar 3.30	Program Animasi Konveyor	35
Gambar 4.1	Cara Cek Koneksi Wifi	37
Gambar 4.2	Status Wifi	38
Gambar 4.3	<i>Wifi Properties</i>	38
Gambar 4.4	<i>Setting IP Address Wifi</i> pada Laptop	39
Gambar 4.5	<i>Command Prompt</i> dengan <i>IP Address Wifi</i>	39
Gambar 4.6	Jarak <i>PC</i> dan Alat dari <i>Google Maps</i>	41
Gambar 4.7	Cek <i>Loss</i> Koneksi Jarak 20 Meter	41
Gambar 4.8	Cek <i>Loss</i> Koneksi Jarak 50 Meter	42
Gambar 4.9	Cek <i>Loss</i> Koneksi Jarak 120 Meter	42
Gambar 4.10	Cek <i>Loss</i> Koneksi Jarak 130 Meter	42
Gambar 4.11	Cek <i>Loss</i> Koneksi Jarak 150 Meter	43
Gambar 4.12	Denah Jarak <i>PC</i> dan Alat	44
Gambar 4.13	Cek <i>Loss</i> Koneksi Jarak 5 Meter	45
Gambar 4.14	Cek <i>Loss</i> Koneksi Jarak 10 Meter	45
Gambar 4.15	Cek <i>Loss</i> Koneksi Jarak 15 Meter	45
Gambar 4.16	Cek <i>Loss</i> Koneksi Jarak 20 Meter	46
Gambar 4.17	Tampilan Awal LabVIEW	47
Gambar 4.18	Barang Tinggi Bergerak di Konveyor 1	48
Gambar 4.19	Barang Tinggi Melewati Konveyor 2	49
Gambar 4.20	Kondisi Ketika Barang Tinggi Dikelompokkan	49
Gambar 4.21	Hasil Penjumlahan Barang Tinggi	49
Gambar 4.22	Barang Sedang Bergerak di Konveyor 1	51
Gambar 4.23	Barang Sedang Naik <i>Lift</i> Melalui Konveyor 2	52
Gambar 4.24	Kondisi Ketika Barang Sedang Dikelompokkan	52
Gambar 4.25	Hasil Penjumlahan Sedang	52
Gambar 4.26	Barang Rendah Bergerak di Konveyor 1	54
Gambar 4.27	Barang Rendah Melewati Konveyor 2	55
Gambar 4.28	<i>Pusher</i> Mendorong Barang Rendah	55
Gambar 4.29	Kondisi Ketika Barang Rendah Dikelompokkan	56
Gambar 4.30	Hasil Penjumlahan Rendah	56

Gambar 4.31	Barang <i>Barcode</i> A Bergerak di Konveyor 1	59
Gambar 4.32	Barang <i>Barcode</i> A Melewati Konveyor 2	59
Gambar 4.33	Kondisi Ketika Barang <i>Barcode</i> A Dikelompokkan	60
Gambar 4.34	Hasil Penjumlahan <i>Barcode</i> A	60
Gambar 4.35	Barang <i>Barcode</i> B Bergerak di Konveyor 1	62
Gambar 4.36	Barang <i>Barcode</i> B Naik <i>Lift</i> Melalui Konveyor 2	63
Gambar 4.37	Kondisi Ketika Barang <i>Barcode</i> B Dikelompokkan	63
Gambar 4.38	Hasil Penjumlahan <i>Barcode</i> B	63
Gambar 4.39	Barang <i>Barcode</i> C Bergerak di Konveyor 1	66
Gambar 4.40	Barang <i>Barcode</i> C Melewati Konveyor 2	66
Gambar 4.41	<i>Pusher</i> Mendorong Barang <i>Barcode</i> C	67
Gambar 4.42	Kondisi Ketika Barang <i>Barcode</i> C Dikelompokkan	67
Gambar 4.43	Hasil Penjumlahan <i>Barcode</i> C	67

-----Halaman ini sengaja dikosongkan-----

DAFTAR TABEL

	HALAMAN
Tabel 2.1 Spesifikasi Arduino Mega 2560	5
Tabel 2.2 Bagian <i>Software</i> Arduino IDE	7
Tabel 3.1 <i>Wiring</i> Modul <i>Relay</i> dan Arduino Mega2560.....	17
Tabel 3.2 <i>Wiring</i> Modul <i>Relay</i> dan <i>Power Supply</i>	18
Tabel 3.3 <i>Wiring</i> Modul <i>Relay</i> dan Motor DC.....	19
Tabel 3.4 <i>Wiring</i> <i>Power Supply</i> dan Motor DC.....	21
Tabel 3.5 <i>Wiring</i> Sensor <i>Infrared</i>	23
Tabel 3.6 <i>Wiring</i> <i>Scanner Barcode</i>	24
Tabel 4.1 Hasil Pengujian Koneksi Tanpa Penghalang	43
Tabel 4.2 Hasil Pengujian Koneksi Dengan Penghalang.....	46
Tabel 4.3 Hasil Pengujian Barang Tinggi.....	50
Tabel 4.4 Hasil Pengujian Barang Sedang.....	53
Tabel 4.5 Hasil Pengujian Barang Rendah	57
Tabel 4.6 Hasil Pengujian <i>Barcode A</i>	61
Tabel 4.7 Hasil Pengujian <i>Barcode B</i>	64
Tabel 4.8 Hasil Pengujian <i>Barcode C</i>	68

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat ini waktu dan biaya transportasi menjadi hal yang sangat dipertimbangkan dalam sebuah perusahaan untuk menekan biaya produksi. Pengangkutan barang baik berupa bahan baku ataupun barang jadi saat ini menjadi yang sangat penting dalam sebuah sistem industri. Karena proses pengelompokkan barang dan memasukkan data yang dilakukan secara manual sering menimbulkan kesalahan perhitungan yang disebabkan oleh *human error*. Maka, data hasil pengelompokkan barang produksi tersebut membutuhkan suatu wadah untuk memuatnya.

Dari masalah tersebut kemudian dirancang sebuah alat pengangkut yang dapat memenuhi tuntutan kondisi di atas, yakni penggunaan konveyor yang bertujuan untuk memindahkan barang yang diproduksi dari suatu tempat ke tempat yang lain serta penggunaan konveyor, *lift*, dan *pusher* untuk proses pengelompokkan. Pengelompokkan pada konveyor banyak dikembangkan dengan tujuan memperbaiki dan meningkatkan kinerja dari sistem. Sistem dari pengelompokkan maupun penyortiran barang menggunakan sensor yaitu sensor *infrared* dan *scanner barcode*. Serta menggunakan komputer untuk menampilkan hasil dari pengelompokkan jumlah barang.

Untuk mengatasi masalah tersebut terciptalah suatu alat yang dapat membantu proses pengelompokkan barang. Proses dari pengelompokkan hanya meletakkan barang di konveyor yang sedang berjalan. Barang akan melewati salah satu sensor aktif yaitu *scanner barcode* atau sensor *infrared* lalu di deteksi. Secara otomatis data hasil dari pengelompokkan barang yang diterima akan dicatat dan ditampilkan pada perangkat komputer atau laptop melalui media *wifi* atau *router*. Sehingga, akan memperkecil *human error* pada industri.

1.2 Permasalahan

Pada Tugas Akhir ini yang menjadi permasalahan yaitu belum tersedia *monitoring* menggunakan perangkat *PC* untuk pengelompokkan dan penghitungan barang pada konveyor dengan menggunakan *wifi*.

1.3 Batasan Masalah

Dari perumusan masalah di atas, maka batasan masalah dari Tugas Akhir ini adalah :

1. Personal komputer diharuskan terkoneksi pada *wifi* agar dapat me-*monitoring* pengelompokkan barang.
2. Pembuatan tampilan menggunakan *software* LabVIEW.
3. *Monitoring* pada alat ini hanya sebatas pemantauan pengelompokkan dan penghitungan barang pada konveyor menggunakan *software* LabVIEW melalui media *wifi* serta tidak mengatur jalannya konveyor maupun pendeteksian barang.

Dengan adanya batasan masalah ini diharapkan hasil akhir atau tujuan dari Tugas Akhir ini dapat dicapai dengan baik.

1.4 Tujuan

Tujuan kami dalam membuat Tugas Akhir ini adalah sebagai berikut :

1. Merancang alat *monitoring* pengelompokkan barang pada konveyor.
2. Membuat alat *monitoring* pengelompokkan pada konveyor.
3. Mengimplementasikan alat *monitoring* pengelompokkan barang pada konveyor agar dapat menghilangkan kemungkinan kesalahan akibat *human error*.

1.5 Sistematika Laporan

Sistematika pembahasan Tugas Akhir ini terdiri dari lima bab, yaitu Pendahuluan, Teori Dasar, Perancangan Sistem, Pengujian dan Analisa, serta Penutup.

Bab I : Pendahuluan

Membahas tentang latar belakang, permasalahan, batasan masalah, maksud dan tujuan, sistematika laporan, serta relevansi.

Bab II : Teori Dasar

Membahas tentang teori - teori penunjang yang diperlukan dan dipergunakan sebagai penunjang pengerjaan Tugas Akhir

Bab III : Perancangan Sistem

Membahas tentang perencanaan dan pembuatan perangkat keras (*hardware*) yang terdiri dari perancangan elektronik dan perancangan mekanik serta pembuatan dan perancangan perangkat lunak (*software*).

Bab IV : Pengukuran dan Analisa

Membahas tentang pengujian dengan cara pengukuran alat yang terdiri dari pengujian pengukuran perangkat keras dan juga perangkat lunak.

Bab V : Penutup

Menjelaskan tentang kesimpulan dari Tugas Akhir ini dan saran-saran untuk pengembangan alat ini lebih lanjut.

1.6 Relevansi

Relevansi yang diharapkan dalam membuat Tugas Akhir ini adalah dari alat yang dibuat dapat berhasil dalam *me-monitoring* pengelompokkan barang pada konveyor melalui media *wifi* dengan perantara *PC* sebagai tampilan hasil data yang diperoleh.

-----Halaman ini sengaja dikosongkan-----

BAB II

TEORI DASAR

2.1 Arduino Mega2560 [1]

Arduino Mega2560 adalah papan *microcontroller* berbasis Atmega 2560. Arduino Mega2560 memiliki 54 pin digital *input / output*. Dimana 15 pin digunakan sebagai *output* PWM, 16 pin sebagai *input analog*, dan 4 pin sebagai *UART (port serial hardware)*, 16 MHz kristal osilator, koneksi USB, *jack power*, *header ICSP*, dan tombol *reset*.

Ini semua yang diperlukan untuk mendukung *microcontroller*. Cukup dengan menghubungkannya ke komputer melalui kabel USB atau *power* dihubungkan dengan adaptor AC – DC atau baterai untuk mulai mengaktifkannya. Bentuk fisik dari Arduino Mega2560 seperti Gambar 2.1 dan spesifikasinya dapat dilihat pada Tabel 2.1.



Gambar 2.1 Arduino Mega2560

Tabel 2.1 Spesifikasi Arduino Mega2560

Spesifikasi	Keterangan
Chip <i>microcontroller</i>	ATmega2560
Tegangan operasi	5V
Tegangan <i>input</i> (yang direkomendasikan, <i>via jack DC</i>)	7V - 12V
Tegangan <i>input</i> (limit, <i>via jack DC</i>)	6V - 20V
Digital I/O pin	54 buah, 6 diantaranya menyediakan PWM output
Analog Input pin	16 buah
Arus DC per pin I/O	20 mA
Arus DC pin 3.3V	50 mA
Memori <i>Flash</i>	256 KB, 8 KB telah digunakan untuk <i>bootloader</i>
SRAM	8 Kb

Spesifikasi	Keterangan
EEPROM	4 Kb
Clock speed	16 Mhz
Dimensi	101.5 mm x 53.4 mm
Berat	37 g

2.2 Ethernet Shield [1]

Ethernet Shield digunakan untuk menambah kemampuan *board* Arduino agar terhubung ke jaringan internet dengan menggunakan kabel RJ45 sebagai koneksinya. *Ethernet shield* berbasis *chip ethernet* Wiznet W5100. *Ethernet library* digunakan dalam menulis program agar *board* Arduino dapat terhubung ke jaringan dengan menggunakan Arduino. Bentuk fisik dari *ethernet shield* seperti Gambar 2.2.



Gambar 2.2 *Ethernet Shield*

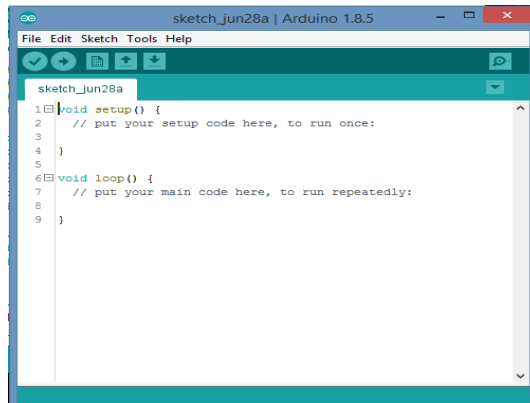
2.3 Sistem Monitoring [1]

Sistem *monitoring* merupakan suatu proses mengumpulkan data dari berbagai sumber barang berupa data yang *real time* dan memberikan informasi kepada operator sistem atau menampilkan data ke perangkat *PC*. Secara garis besar tahapan dalam sebuah sistem *monitoring* terbagi ke dalam tiga tahapan, yaitu proses di dalam pengumpulan data *monitoring*, proses di dalam analisis data *monitoring* dan proses menampilkan data hasil *monitoring*.

2.4 Arduino IDE [2]







IDE merupakan kependekan dari *Integrated Development Environment*. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Sebelum dijual ke pasaran, *IC microcontroller* Arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler. Arduino IDE dibuat dari bahasa pemrograman JAVA serta dilengkapi dengan library C/C++ dapat dilihat seperti

Gambar 2.3. Dari tampilan pada *software* Arduino IDE terdapat beberapa gambar yang dapat dijelaskan pada Tabel 2.2.



Gambar 2.3 *Software* Arduino IDE

Tabel 2.2 Bagian *Software* Arduino IDE

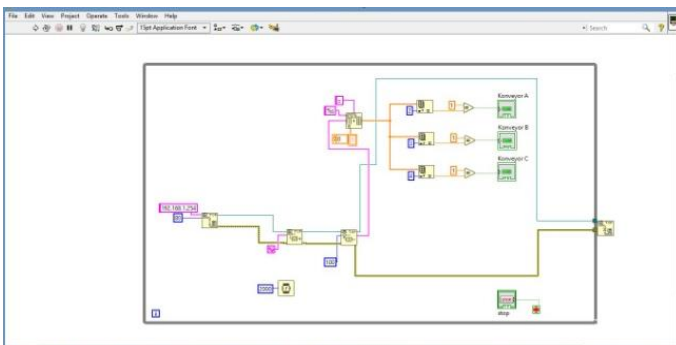
Gambar	Pengertian
	<i>Verify</i> melakukan pengecekan kode yang dibuat apakah sudah sesuai dengan kaidah pemrograman yang ada atau belum
	<i>Upload</i> melakukan kompilasi program atau kode yang kamu buat menjadi bahasa yang dapat dipahami oleh Arduino
	<i>New</i> berfungsi untuk membuat <i>sketch</i> baru
	<i>Open</i> berfungsi untuk membuka <i>sketch</i> yang pernah kamu buat dan membuka kembali untuk dilakukan <i>editing</i> atau sekedar <i>upload</i> ulang ke Arduino.
	<i>Save</i> berfungsi untuk menyimpan <i>sketch</i> yang telah dibuat.
	<i>Serial Monitor</i> berfungsi untuk menampilkan data apa saja yang dikirimkan. <i>Serial monitor</i> ini sangat berguna sekali ketika ingin membuat program atau melakukan <i>debugging</i> tanpa menggunakan LCD pada Arduino.

Gambar	Pengertian
	<i>Serial monitor</i> ini dapat digunakan untuk menampilkan nilai proses, nilai pembacaan dan sebagainya.

2.5 Software LabVIEW [2]

LabVIEW singkatan dari *Laboratory Virtual Instrumentation Engineering Workbench* adalah suatu bahasa pemrograman yang menggunakan *icon* sebagai ganti bentuk teks untuk menciptakan aplikasi dalam pemrosesan dan menyampaikan hasil data. LabVIEW pertama kali dikembangkan oleh perusahaan *National Instruments* pada tahun 1986. Perangkat lunak ini dapat dijalankan pada sistem operasi *Linux*, *Unix*, *Mac OS X* dan *Windows*.

Karena menggunakan blok diagram pada *software* LabVIEW sebagai bahasa pemrogramannya ketika *error* terjadi pun dapat dilihat di mana letak terjadinya kesalahan. Contoh blok diagram yang ada di LabVIEW dapat dilihat pada Gambar 2.4.



Gambar 2.4 Blok Diagram *Software* LabVIEW

2.6 Konveyor [3]

Macam-macam konveyor yang biasa digunakan ada beberapa macam yaitu :

2.6.1 Belt Conveyor

Belt conveyor atau yang disebut dengan ban berjalan merupakan alat yang dapat memudahkan untuk memindahkan material baik berupa *Unit Load* atau *Bulk Material* dari tempat ke suatu tempat lainnya secara terus menerus dengan kapasitas yang diperlukan. *Unit Load* adalah

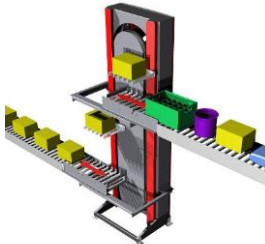
benda berupa barang misalnya kotak, kantong, balok, dan lain-lain yang berisikan makanan, minuman, atau sejenisnya. Sedangkan untuk *Bulk Material* berupa butiran, bubuk atau serbuk, misalnya pasir dan semen. Mesin ini dapat digunakan dalam industri pertambangan, metalurgi dan batu bara, men-*transfer* pasir, material besar, atau material dalam kemasan berupa makanan atau minuman. *Belt conveyor* dapat dipasang secara *horizontal* atau tertidur untuk memenuhi kebutuhan yang diperlukan dalam memindahkan material seperti Gambar 2.5.



Gambar 2.5 *Belt Conveyor*

2.6.2 *Vertical Conveyor*

Vertical conveyor merupakan alat yang digunakan untuk memindahkan barang bahan produksi atau semacamnya dengan berat tertentu sesuai kebutuhan yang diperlukan. Untuk memindahkannya dengan cara *vertical* yang berbentuk *lift* guna menaikkan dan menurunkan suatu barang untuk menyortirnya seperti pada Gambar 2.6.



Gambar 2.6 *Vertical Conveyor*

2.6.3 *Plastic Flain Chain Conveyor*

Conveyor jenis ini banyak ditemukan dalam dunia industri makanan dan minuman berupa kemasan. Material kontakannya terbuat dari bahan *non* logam yaitu nilon, plastik, PU. *Conveyor* ini memiliki keunggulan pada sistemnya yaitu permukaan atas *conveyor* rata atau *flat*

dan dapat bergerak melengkung. Sehingga efektif untuk digunakan apabila area kerja memiliki tempat yang sedikit, sedangkan proses memerlukan *conveyor* dalam jarak yang cukup panjang untuk mengantisipasi antrian produk. Hal ini dapat dilihat pada Gambar 2.7



Gambar 2.7 *Plastic Flat Chain Conveyor*

2.6.4 Vibrating Conveyor

Untuk menggerakkan barang pada *vibrating conveyor* menggunakan efek getaran secara *horizontal* atau *vertical* sesuai dengan kebutuhan yang diperlukan seperti Gambar 2.8 dan Gambar 2.9.



Gambar 2.8 *Horizontal Vibrating Conveyor*



Gambar 2.9 *Vertical Vibrating Conveyor*

2.6.5 Hanging atau Overhead Conveyor

Alat *conveyor* ini dapat memindahkan barang dengan cara digantung. Sedangkan untuk tenaga penggeraknya menggunakan gravitasi atau motor. Bentuk *conveyor* ini dapat dilihat pada Gambar 2.10.



Gambar 2.10 *Hanging* atau *Overhead Conveyor*

2.7 Motor DC [4]

Motor DC atau *DC motor* adalah suatu perangkat yang mengubah energi listrik menjadi energi kinetik atau gerakan. Motor DC disebut sebagai motor arus searah. Motor DC ini menghasilkan putaran per menit yang dikenal dengan istilah RPM (*Revolutions Per Minute*) dan dapat dibuat berputar searah jarum jam maupun berlawanan arah jarum jam jika polaritas listrik yang diberikan pada motor DC tersebut dibalik. Terdapat dua bagian utama pada sebuah motor DC, yaitu *stator* dan *rotor*. *Stator* adalah bagian motor yang tidak berputar, bagian yang statis ini terdiri dari rangka dan kumparan medan. Sedangkan *rotor* adalah bagian yang berputar, bagian *rotor* ini terdiri dari kumparan jangkar. Motor DC dengan tegangan 21,6V serta arus sebesar 0,6A dan RPM sebanyak 70 kali putaran dapat dilihat pada Gambar 2.11.



Gambar 2.11 Motor DC

2.8 Relay [5]

Relay adalah komponen elektronika berupa saklar atau *switch* elektrik yang dioperasikan menggunakan listrik. *Relay* juga disebut sebagai komponen *electromechanical* atau elektromekanikal yang terdiri dari dua bagian utama yaitu *coil* atau elektromagnet dan kontak saklar atau mekanikal. Komponen *relay* menggunakan prinsip elektromagnetik sebagai penggerak kontak saklar, sehingga dengan menggunakan arus

listrik yang kecil atau *low power*, dapat menghantarkan arus listrik yang memiliki tegangan lebih tinggi.

Penggunaan *relay* perlu diperhatikan tegangan pengontrolnya serta kekuatan *relay* men-*switch* arus atau tegangan. Ukurannya tertera pada *body relay* misalnya *relay* 12 VDC / 4A 220V, artinya tegangan yang diperlukan sebagai pengontrol adalah 12 VDC dan mampu men-*switch* arus listrik (maksimal) sebesar 4A pada tegangan 220V. Contoh *relay* yang umum digunakan dapat dilihat pada Gambar 2.12.



Gambar 2.12 *Relay*

Sedangkan modul *relay* seperti Gambar 2.13 dapat digunakan sebagai *switch* untuk menjalankan berbagai peralatan elektronik. Misalnya lampu listrik, motor listrik, dan berbagai peralatan elektronik lainnya. Untuk kendali *ON* atau *OFF* *switch* (*relay*), sepenuhnya ditentukan oleh nilai *output* sensor, yang telah diproses mikrokontroller dan akan menghasilkan perintah kepada *relay* untuk melakukan fungsi *ON* atau *OFF*. Penggunaan dari modul *relay* sepenuhnya menggunakan program.



Gambar 2.13 Modul *Relay*

BAB III

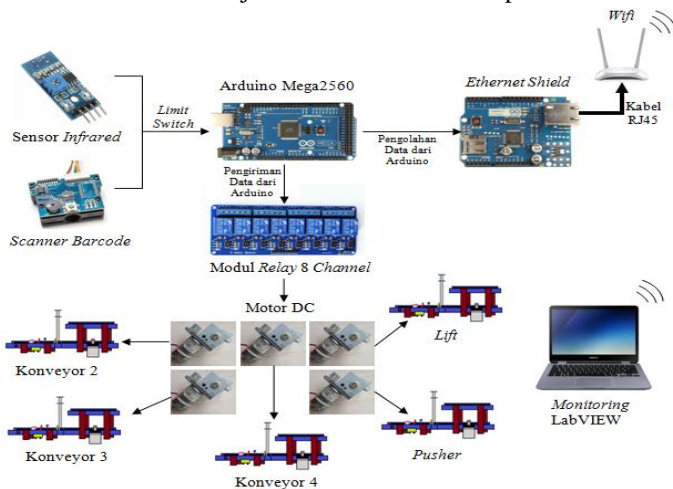
PERANCANGAN SISTEM

Pada Tugas Akhir membuat alat pengelompokkan barang menggunakan sensor *infrared* dan *scanner barcode*, ada beberapa tahapan yang dilakukan yaitu :

1. Diagram Fungsional Alat
2. Perancangan Perangkat Keras (*Hardware*)
 - a. Perancangan Konveyor
 - b. Perancangan Sistem *Monitoring*
3. Perancangan Elektronik
 - a. Perancangan Rangkaian *Ethernet Shield*
 - b. Perancangan Keseluruhan
4. Perancangan Perangkat Lunak (*Software*)
 - a. *Flowchart*
 - b. *Interface Ethernet Shield* untuk Arduino
 - c. Langkah-Langkah Pemograman LabVIEW

3.1 Diagram Fungsional Alat

Perancangan sistem dalam pembuatan alat ini secara garis besar disertai urutan dan cara kerja alat ini di ilustrasikan pada Gambar 3.1.



Gambar 3.1 Diagram Fungsional Alat

Perancangan sistem dalam pembuatan alat ini secara garis besar disertai urutan dan cara kerja di ilustrasikan pada Gambar 3.1. Terdapat dua macam sensor yang digunakan yaitu sensor *infrared* untuk mendeteksi barang berdasarkan ketinggian, sedangkan *scanner barcode* digunakan untuk mendeteksi kode batang. Dapat dijelaskan ketika barang melewati salah satu sensor dan dideteksi, nilai hasil data yang diperoleh masuk ke dalam Arduino dan mengirim data ke *relay* agar dapat menjalankan konveyor. Motor DC digunakan sebagai penggerak konveyor 2, konveyor 3, konveyor 4, *lift*, dan *pusher* sesuai barang yang akan di kelompokkan. Motor ini bergerak berdasarkan *relay* pengaturnya. Terkecuali konveyor 1, dikarenakan motor DC untuk konveyor 1 akan aktif secara otomatis jika tombol *power* dalam posisi *ON*.

Arduino Mega2560, berfungsi sebagai pengontrol perangkat elektronik dan dapat menyimpan program didalamnya. *Ethernet shield* berfungsi sebagai penghubung antara Arduino dengan personal komputer tanpa harus menggunakan kabel serial Arduino yang jaraknya terbatas, sehingga dengan menggunakan *ethernet shield* ini memungkinkan Arduino tetap terhubung dengan personal komputer dengan jarak yang lebih jauh dibandingkan dengan menggunakan kabel serial Arduino yaitu dengan menggunakan *wifi* dan dihubungkan oleh kabel RJ45. Hasil pengelompokkan barang yang di deteksi akan ditampilkan pada personal komputer dengan menggunakan *software* LabVIEW melalui media *wifi*.

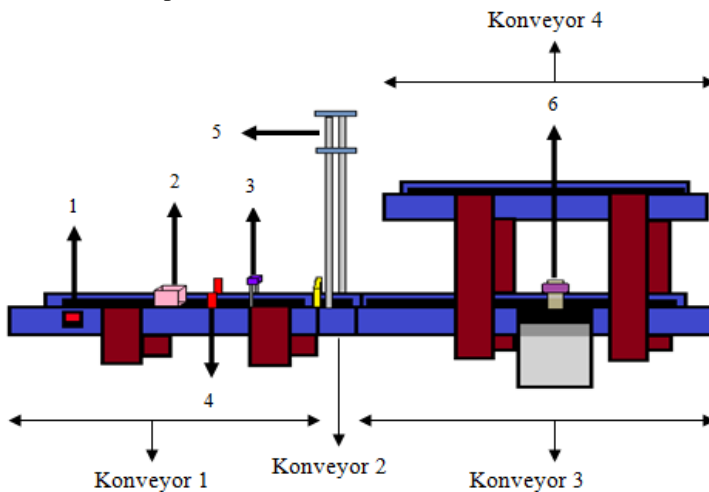
3.2 Perancangan Perangkat Keras (*Hardware*)

Perancangan Perangkat keras ini dimaksudkan agar dalam pelaksanaan pengerjaan Tugas Akhir menjadi lebih mudah dan cepat karena setiap detail perangkat keras (*hardware*) yang akan dikerjakan baik dari sisi elektronik maupun mekanik telah direncanakan secara terperinci dan baik.

3.2.1 Perancangan Konveyor

Pada perancangan kali ini menggunakan dua sensor untuk mendeteksi barang yang terdapat pada konveyor yaitu sensor *infrared* dan *scanner barcode*. Sensor *infrared* bekerja apabila barang dengan dimensi ukuran tertentu akan di deteksi, begitu pula *scanner barcode* akan aktif jika barang dengan *barcode* di deteksi. Motor DC digunakan menggerakkan konveyor, *lift*, dan *pusher* dengan menggunakan *relay*

sebagai pengendali untuk melakukan pengelompokkan. Rancangan konveyor yang akan digunakan terdiri dari konveyor 1, konveyor 2, konveyor 3, konveyor 4, *lift*, dan *pusher*. Konveyor 1, 2, 3, dan *pusher* berada di lantai bawah, sedangkan konveyor 4 berada di lantai atas. *Lift* konveyor bekerja sesuai posisi barang yang telah ditentukan secara keatas ataupun tidak. Sedangkan *pusher* bergerak maju atau tidak. Untuk konveyor 1 akan bergerak ketika tombol *power* ditekan dikarenakan mendapatkan sumber 8V secara langsung. Desain alat yang dibuat semirip mungkin dengan aslinya untuk memudahkan pada saat pembuatan alat seperti Gambar 3.2.



Gambar 3.2 Rancangan Konveyor Serta *Lift*

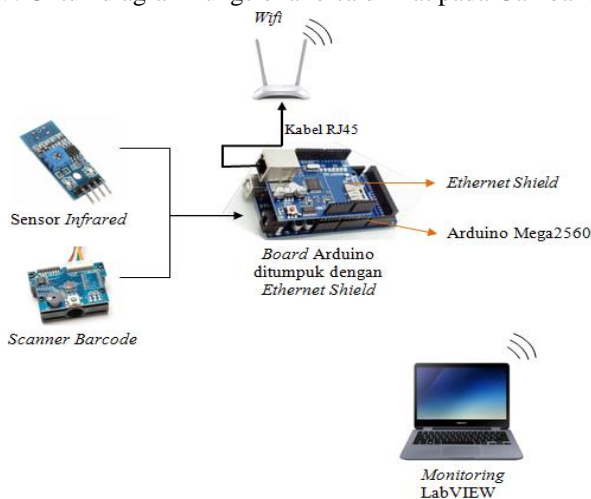
Keterangan :

1. Tombol *Power*
2. Barang
3. *Scanner Barcode*
4. *Sensor Inframerah*
5. *Lift* Konveyor
6. *Pusher*

3.2.2 Perancangan Sistem *Monitoring*

Perancangan ini akan dijelaskan melalui diagram yang mewakili dari perangkat keras pada Gambar 3.3 dengan fungsi tersendiri. Dari alat

yang digunakan ketika tombol *power* ditekan, secara otomatis konveyor 1 akan bergerak. Ketika barang telah terdeteksi oleh sensor maka hasil data yang diperoleh akan dikirimkan ke Arduino untuk diolah dan memberikan perintah kepada konveyor, *lift*, dan *pusher* dalam pendistribusian barang. Disaat pendistribusian, hasil data yang diperoleh dari deteksi barang dikirimkan ke *ethernet shield* dan melakukan proses *monitoring*. *Ethernet shield* berfungsi sebagai pengirim data ke laptop melalui *wifi*, hasil data yang diterima dari Arduino akan ditampilkan pada laptop untuk mengetahui hasilnya dengan menggunakan *software* LabVIEW. Untuk diagram fungsional bisa dilihat pada Gambar 3.3.



Gambar 3.3 Diagram Fungsional Sistem *Monitoring*

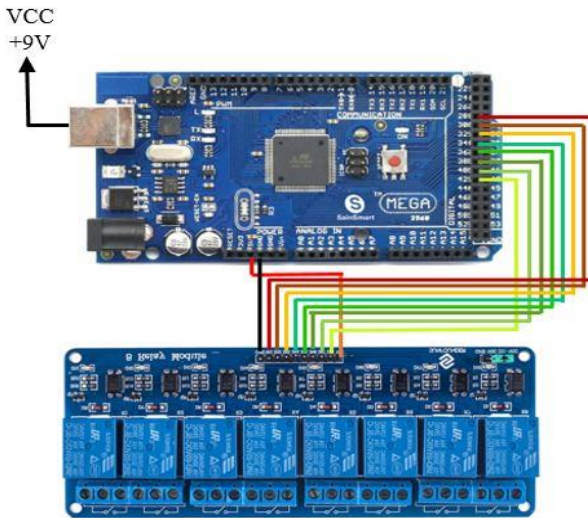
3.3 Perancangan Elektronik

Pada perancangan elektronik kali ini yang digunakan pada pembuatan sistem agar alat dapat berjalan dengan baik. Adapun tahapan-tahapan yang dilakukan yaitu :

3.3.1 Wiring Modul *Relay* dan Arduino Mega2560

Pada alat ini menggunakan *relay module 8 channel* dengan *input* 5V dan *output* 12V. *Relay* digunakan untuk menyuplai motor pada konveyor yang berfungsi sebagai mengatur *ON* atau *OFF* pada motor. Kerja dari *relay* tersebut diatur oleh program Arduino. Sedangkan Arduino mendapatkan sumber sebesar 9V. Untuk lebih jelasnya, dapat

dilihat pada Gambar 3.4 dari *wiring* modul *relay* dan Arduino Mega2560 dan Tabel 3.1.



Gambar 3.4 *Wiring* Modul *Relay* dan Arduino Mega2560

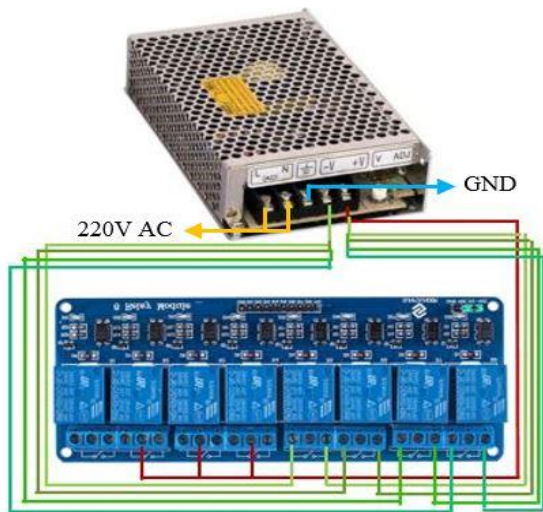
Tabel 3.1 *Wiring* Modul *Relay* dan Arduino Mega2560

Pin Modul Relay	Pin Arduino Mega2560
GND	GND
VCC	+5V
IN1	28
IN2	30
IN3	32
IN4	34
IN5	36
IN6	38
IN7	40
IN8	42

3.3.2 *Wiring* Modul *Relay* dan *Power Supply*

Pada tahap ini, menggunakan *relay module 8 channel* serta *power supply* 12V 5A. Motor dapat bergerak berdasarkan *relay* sebagai pengaturnya. Agar motor dapat bergerak maka *relay* harus mendapatkan suplai tegangan sesuai kebutuhan motor. Suplai tegangan tersebut

berasal dari *power supply*. Agar *relay* dan *power supply* aktif maka *power supply* harus mendapatkan sumber sebesar 220V AC. Berikut Gambar 3.5 wiring *relay* dan *power supply* serta Tabel 3.2



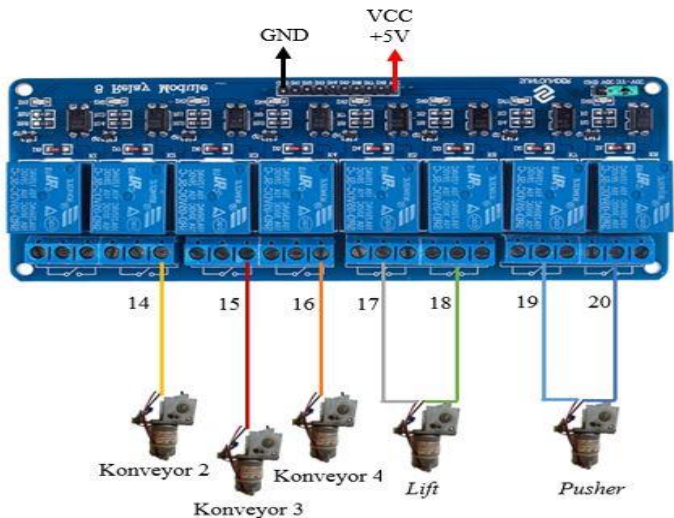
Gambar 3.5 Wiring Modul Relay dan Power Supply

Tabel 3.2 Wiring Modul Relay dan Power Supply

Power Supply	Modul Relay		
	NO	NC	COM
+12 V	Relay 5	-	-
	Relay 6	-	-
	Relay 7	-	-
	Relay 8	-	-
	-	-	Relay 2
	-	-	Relay 3
-12 V	-	Relay 5	-
	-	Relay 6	-
	-	Relay 7	-
	-	Relay 8	-

3.3.3 Wiring Modul Relay dan Motor DC

Setelah penyambungan antara *relay* dan *power supply* berhasil. Selanjutnya adalah penyambungan modul *relay* dan motor DC. Modul *relay* sendiri mendapatkan sumber sebesar 5V yang berasal dari Arduino, untuk *wiring* modul *relay* dan Arduino dapat dilihat pada Gambar 3.4. Agar disaat program dari Arduino berfungsi, maka otomatis *relay* yang tersambung dengan motor akan aktif, kemudian motor bergerak sesuai program yang ditentukan. Alat ini menggunakan beberapa motor DC dalam menggerakkan konveyor. Untuk *wiring relay* dan motor DC dapat dilihat pada Gambar 3.7 serta Tabel 3.3.



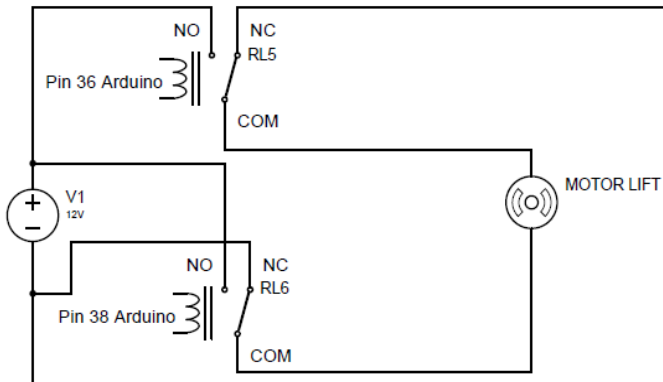
Gambar 3.6 Wiring Modul Relay dan Motor DC

Tabel 3.3 Wiring Modul Relay dan Motor DC

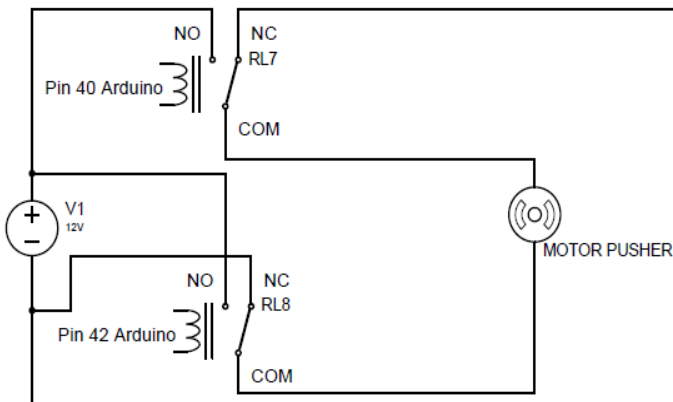
Pin Modul Relay	Motor	Keterangan
NC Relay 2	Motor 14 (+)	Menggerakkan Konveyor 2
NC Relay 3	Motor 15 (+)	Menggerakkan Konveyor 3
NC Relay 4	Motor 16 (+)	Menggerakkan Konveyor 4
COM Relay 5	Motor 17 (+)	Menggerakkan Lift Naik
COM Relay 6	Motor 18 (+)	Menggerakkan Lift Turun
COM Relay 7	Motor 19 (+)	Menggerakkan Pusher Untuk Mendorong Barang
COM Relay 8	Motor 20 (+)	Menggerakkan Pusher Ke Posisi

Pin Modul Relay	Motor	Keterangan
		Awal

Agar *lift* dan *pusher* dapat berputar secara dua arah yaitu dengan menggunakan dua *relay* yang berbeda. Kontak *NC relay* tersambung pada sumber tegangan sebesar -12V sedangkan *NO relay* tersambung dengan sumber tegangan +12V. *Commond relay* tersambung dengan motor DC. Untuk lebih jelasnya dapat dilihat pada Gambar 3.7 dan Gambar 3.8.



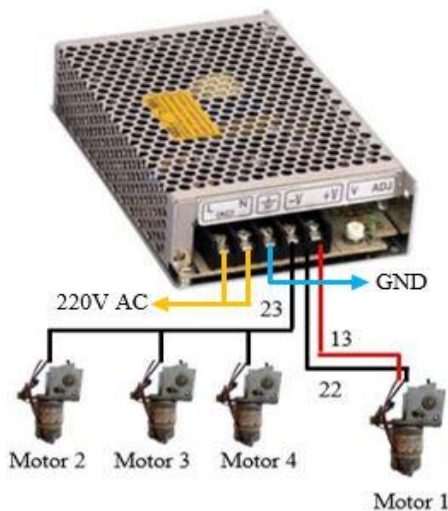
Gambar 3.7 Rangkaian Motor Penggerak *Lift*



Gambar 3.8 Rangkaian Motor Penggerak *Pusher*

3.3.4 *Wiring Power Supply dan Motor DC*

Dalam menggerakkan konveyor menggunakan motor DC sebagai komponen utamanya. Setiap masing-masing konveyor, *lift*, dan *pusher* dijalankan oleh satu motor DC yang berbeda. Namun *lift* dan *pusher* memiliki dua gerakan yang berbeda yaitu *lift* bekerja secara naik turun dan *pusher* bekerja secara maju mundur. Dari dua gerakan tersebut setiap *lift* dan *pusher* dijalankan oleh satu motor namun menggunakan dua *relay* untuk *wiring*-nya dapat dilihat pada Gambar 3.6. Sedangkan motor DC pada konveyor 1 langsung terhubung dengan *power supply*. Untuk menggerakkan motor DC harus mendapatkan sebuah sumber yaitu *power supply*. Agar *power supply* dapat aktif dan motor DC dapat bergerak maka harus diberi sumber tegangan sebesar 220V AC. Berikut adalah Gambar 3.9 dari *wiring power supply* dan motor DC. Untuk lebih jelasnya mengenai *wiring power supply* dan motor DC dapat dilihat pada Tabel 3.4.



Gambar 3.9 *Wiring Power Supply dan Motor DC*

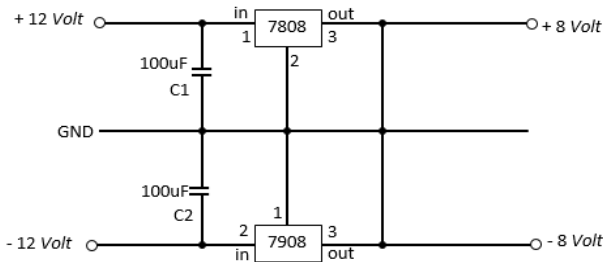
Tabel 3.4 *Wiring Power Supply dan Motor DC*

<i>Power Supply</i>	<i>Wiring</i>	<i>Motor</i>
-12V	23	Motor 2 Untuk Menggerakkan Konveyor 2
		Motor 3 Untuk Menggerakkan Konveyor 3
		Motor 4 Untuk Menggerakkan Konveyor 4

Power Supply	Wiring	Motor
-12V	22	Motor 1 Untuk Menggerakkan Konveyor 1

3.3.5 Rangkaian *Voltage Regulator*

Pada alat ini menggunakan rangkaian *voltage regulator* yaitu LM7808 sebagai *output* positif dan LM7908 sebagai *output* negatif. Rangkaian ini digunakan untuk mensuplai motor konveyor 1 dikarenakan pada suplai 12V konveyor bergerak terlalu cepat sehingga sensor susah untuk mendeteksi barang. Maka dari itu, suplai 12V diturunkan menjadi 8V. Berikut adalah rangkaian *voltage regulator* yang digunakan pada Gambar 3.10.



Gambar 3.10 Rangkaian *Voltage Regulator*

3.3.6 Perancangan *Ethernet Shield*

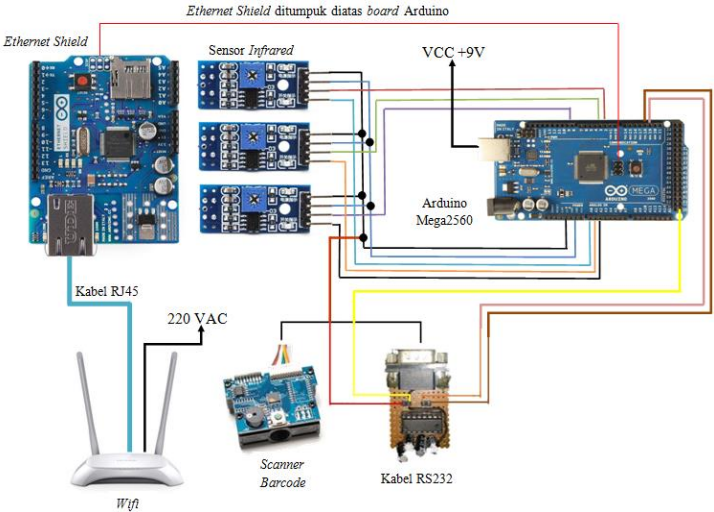
Arduino Mega2560 tidak menyediakan perangkat bawaan untuk terhubung dengan jaringan *wireless*. Arduino *ethernet shield* merupakan modul tambahan yang memungkinkan Arduino bisa terhubung dengan *network*, sehingga Arduino dapat mengirim hasil data ke laptop. Untuk pemasangannya cukup mudah yaitu dengan ditumpuk diatas papan atau dengan menancapkan tiap-tiap pin ke *board* Arduino Mega2560 seperti Gambar 3.11.



Gambar 3.11 Rancangan *Ethernet Shield*

3.3.7 Wiring Ethernet Shield, Sensor, dan Wifi

Perancangan ini meliputi *wiring* atau pengkabelan. *Wiring* kali ini meliputi Arduino, *ethernet shield*, sensor *infrared*, *scanner barcode* dan *wifi*. Dari Gambar 3.12 *ethernet shield* ditancapkan ke Arduino, sedangkan sensor disambungkan ke pin yang telah ditentukan, untuk *scanner barcode* dihubungkan terlebih dahulu dengan kabel RS232. *Wifi* dihubungkan dengan *ethernet shield* melalui kabel RJ45 guna mengirim hasil data ke perangkat komputer. Agar Arduino, *ethernet shield*, sensor *infrared*, *scanner barcode* aktif maka Arduino dihubungkan dengan sumber 9V sedangkan *wifi* aktif apabila diberi sumber 220 VAC. Untuk mengetahui lebih jelas mengenai *wiring* dari sensor *infrared* dan ke *board* Arduino dapat dilihat pada Tabel 3.4.



Gambar 3.12 Wiring Ethernet Shield, Sensor dan Wifi

Tabel 3.5 Wiring Sensor Infrared

Pin Sensor	Pin Arduino
Vcc Sensor Tinggi	5V
Vcc Sensor Sedang	
Vcc Sensor Rendah	
Gnd Sensor Tinggi	GND
Gnd Sensor Sedang	
Gnd Sensor Rendah	

Pin Sensor	Pin Arduino
DO Sensor Tinggi	2
DO Sensor Sedang	3
DO Sensor Rendah	8
A0 Sensor Tinggi	A0
A0 Sensor Sedang	A1
A0 Sensor Rendah	A2

Sedangkan mengenai *wiring* dari *scanner barcode* langsung terhubung dengan kabel RS232 kemudian dihubungkan Arduino dengan penjelasan pada Tabel 3.5.

Tabel 3.6 *Wiring Scanner Barcode*

IC Max 232	Sambungkan Ke
Pin 11(T1 <i>Input</i>)	Pin 19 (TX) Arduino
Pin 12 (R1 <i>Output</i>)	Pin 18 (RX) Arduino
Pin 13 (T1 <i>Output</i>)	Pin 3 (RXD) RS232
Pin 14 (R1 <i>Input</i>)	Pin 2 (TXD) RS232
Pin 15 (<i>GND</i> MAX232)	Pin 5 (<i>GND</i> RS232)
Pin 16 (<i>Vcc</i> MAX232)	<i>Vcc</i> Arduino
	Pin 5 (<i>GND</i>) RS232 ke <i>GND</i> Arduino

3.4 Perancangan Perangkat Lunak (*Software*)

Pada Tugas Akhir kali ini untuk perancangan *software* diperlukan beberapa tahapan yang harus dilakukan terlebih dahulu. Tahapan tersebut adalah membuat algoritma atau suatu pemrograman dari alat yang sudah dijalankan setelah membuat *flowchart* dari algoritma tersebut agar alat terlebih lebih sederhana. Setelah itu barulah memprogram fungsi terkait pada *software* Arduino.

3.4.1 *Flowchart*

Flowchart pada alat ini merupakan *flowchart* sistem kerja dari pengelompokkan barang serta *monitoring*. Pada *flowchart* sistem kerja pada Gambar 3.13 menggunakan sensor *infrared* dan *scanner barcode*. Sensor *infrared* digunakan untuk mendeteksi barang berdasarkan barang tinggi, barang sedang, dan barang rendah. Sedangkan *scanner barcode* sebagai pendeteksi barang berdasarkan *barcode* A, *barcode* B, dan *barcode* C. Selain itu, *relay* juga perlu digunakan dalam proses

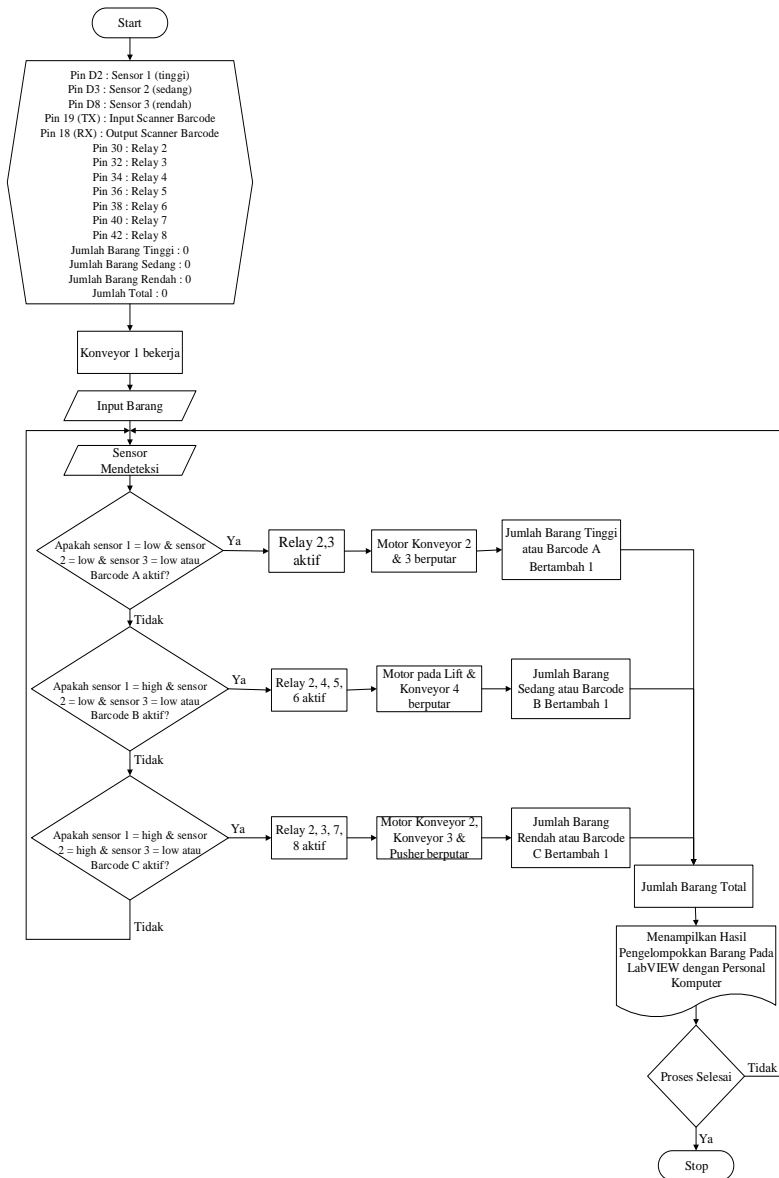
pengelompokkan dan perlu di inisialisasi terlebih dahulu. Kondisi awal pada jumlah masing-masing barang dan jumlah total adalah 0. Disaat konveyor 1 berjalan maka sensor mulai mendeteksi sehingga sistem akan bekerja untuk mengidentifikasi barang apa yang lewat. Pengidentifikasian barang menggunakan sensor *infrared* dan *scanner barcode*.

Sensor *infrared* akan mendeteksi barang tinggi apabila ketiga sensor aktif *low*. Kemudian *relay* 2 dan 3 aktif dan mengontak motor guna menggerakkan konveyor 2 dan konveyor 3. Setelah selesai jumlah pada barang tinggi ialah satu dan jumlah total bertambah satu. Begitu pula dengan *scanner barcode* yang mendeteksi barang *barcode* A. *Relay* 2 serta 3 aktif dan mengontak motor untuk menggerakkan konveyor 2 dan konveyor 3. Setelah selesai jumlah pada barang *barcode* A berjumlah satu dan jumlah total menjadi dua.

Jika sensor *infrared* mendeteksi barang sedang apabila sensor 2 dan sensor 3 aktif *low*. Kemudian *relay* 2, 4, 5, dan 6 aktif dan mengontak motor guna menggerakkan konveyor 4 dan *lift*. Setelah selesai jumlah pada barang sedang berjumlah satu dan jumlah total bertambah sesuai dengan barang yang di identifikasi. Begitu pula dengan *scanner barcode* yang mendeteksi barang *barcode* B. *Relay* 2, 4, 5, dan 6 aktif dan mengontak motor untuk menggerakkan konveyor 4 dan *lift*. Setelah selesai jumlah pada barang *barcode* B berjumlah satu dan jumlah total bertambah sesuai dengan pengidentifikasiannya terhadap barang.

Tahap terakhir sensor *infrared* mendeteksi barang rendah apabila sensor 3 aktif *low*. Kemudian *relay* 2, 3, 7, dan 8 aktif dan mengontak motor guna menggerakkan konveyor 2, konveyor 3, dan *pusher*. Setelah selesai jumlah pada barang rendah berjumlah satu dan jumlah total bertambah sesuai dengan barang yang di identifikasi. Begitu pula dengan *scanner barcode* yang mendeteksi barang *barcode* C. *Relay* 2, 3, 7, dan 8 aktif dan mengontak motor untuk menggerakkan konveyor 2, konveyor 3, dan *pusher*. Setelah selesai jumlah pada barang *barcode* C berjumlah satu dan jumlah total bertambah sesuai dengan pengidentifikasiannya terhadap barang.

Setelah barang di deteksi oleh sensor kemudian hasil dari pengelompokkan barang maupun penjumlahan juga akan di *monitoring* pada perangkat komputer dengan menggunakan LabVIEW. Hal ini dapat dilihat pada Gambar 3.13.



Gambar 3.13 Flowchart Sistem Kerja Pengelompokan dan *Monitoring*

3.4.2 Penyusunan *Ethernet Shield* untuk Arduino

Untuk perancangan *ethernet shield* ke Arduino, *software* yang digunakan adalah bernama Arduino IDE. Memprogram Arduino juga harus dilakukan beberapa tahapan, yaitu untuk pemrograman *software* Arduino harus di *setting* terlebih dahulu *board* Arduinonya agar penggunaan Arduino cocok. Dalam purwarupa kali ini Arduino menggunakan Arduino Mega2560. Untuk *setting board* Arduino bisa masuk ke *tools - board* setelah itu pilihlah *board* Arduino yang digunakan yaitu Arduino Mega2560. Selanjutnya me-*setting serial*-nya. *Serial* sendiri merupakan kabel Arduino yang dihubungkan kepada komputer atau laptop. *Setting serial* bisa masuk *tools - serial* lalu pilih COM yang sesuai dengan Arduino yang terpasang.

Karena kita menggunakan *ethernet shield* sebagai media komunikasi, maka kita harus melakukan *setting* untuk mengatur alamat *IP address ethernet shield* yang digunakan. Selain itu melakukan program pengalaman pada sensor *infrared*, *scanner* barcode, *relay* serta program pengiriman hasil data ke *ethernet shield*.

a. Inisialisasi Pin Alamat *IP Address Ethernet Shield*

```
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};  
IPAddress ip(192, 168, 0, 175);  
EthernetServer server(80);
```

Gambar 3.14 Alamat *IP Address Ethernet Shield*

Pada Gambar 3.14 terlihat alamat *IP* yang digunakan. Alamat *IP* yang digunakan yaitu 192.168.0.175 dan *port* 80.

b. Program Inisialisasi Sensor *Infrared*

Pada Gambar 3.15 terlihat program pengalaman sensor *infrared* untuk mengirim data pembacaan ke *ethernet shield*.

```
const int DsensorTinggi = 2;  
const int DsensorSedang = 3;  
const int DsensorRendah = 8;  
const int AsensorTinggi = A0;  
const int AsensorSedang = A1;  
const int AsensorRendah = A2;
```

Gambar 3.15 Inisialisasi Sensor *Infrared*

c. Program Inisialisasi *Scanner Barcode*

Pada Gambar 3.16 terlihat program pengalaman *scanner barcode* untuk mengirim data pembacaan ke *ethernet shield*.

```
String inputString = "";  
boolean stringComplete = false;  
String barcode;
```

Gambar 3.16 Inisialisasi *Scanner Barcode*

Dalam perancangan program pada *software* Arduino dengan fungsi terkait Gambar 3.14 untuk *me-setting* alamat *IP* yang digunakan untuk menambah kemampuan *board* Arduino agar terhubung ke jaringan komputer. agar program dapat berjalan sesuai yang dikehendaki. Sedangkan inisialisasi sensor *infrared* digunakan agar nilai yang diinginkan dapat membuat program menghasilkan data yang *real* sesuai dengan hasil yang di deteksi, hal ini dapat dilihat pada Gambar 3.14. Begitu pula dengan *scanner barcode* dapat dilihat pada Gambar 3.15.

d. Program Inisialisasi *Relay*

```
#define relayon 0  
#define relayoff 1  
  
int relay1 = 5;  
int relay2 = 6;  
int relay3 = 7;  
int relay4 = 8;  
int relay5 = 9;  
int relay6 = 10;  
int relay7 = 11;  
int relay8 = 12;
```

Gambar 3.17 Inisialisasi *Relay*

Pada Gambar 3.17 merupakan program inisialisasi *relay* dengan pin Arduino yang digunakan adalah pin 5 hingga pin 12. Inisialisasi dari masing-masing pin *relay* dari Arduino digunakan untuk program pengelompokkan.

e. Program Pengiriman Data Ke *Ethernet Shield*

```
proses_decision();  
send2ethernet();  
tampil_lcd();  
  
if(flag_doProses>=100) //3dtk 60*50ms  
{  
    proses(do_proses);  
    do_proses=0;  
    flag_set=0;  
    flag_doProses=0;  
    Serial.print("proses ");  
    Serial.print(do_proses);  
    Serial.println("is done");  
}
```

Gambar 3.18 Pengiriman Data Ke *Ethernet Shield*

Dari Gambar 3.18 merupakan program pengiriman hasil data ke *ethernet shield* yang diperoleh dalam pendeteksian barang oleh sensor. Program tersebut digunakan agar dapat menampilkan hasil data dengan perangkat personal komputer menggunakan LabVIEW melalui media *wifi*.

3.5 Langkah-Langkah Pemograman LabVIEW

Setelah melakukan *programming* dan pengiriman data dari sensor *infrared* dan *scanner barcode* menggunakan *software* Arduino, maka perlu dilakukan pengaturan program menggunakan *software* LabVIEW. Program pada LabVIEW digunakan untuk membaca data yang dikirimkan oleh *ethernet shield* melalui perantara *wifi*.

Proses pengambilan data yang dikirim oleh *ethernet shield* akan berjalan apabila terdapat kesesuaian antara *IP address* dan *wifi* yang telah di *upload* ke *ethernet shield* dengan *IP address* dan *wifi* yang dimasukkan lewat *software* LabVIEW. Setelah hasil data yang dikirimkan oleh sensor *infrared* dan *scanner barcode* lewat *ethernet shield* didapatkan oleh komputer, maka *software* LabVIEW akan mengambil data pemilah atau mengelompokkan barang. Hasil data yang telah didapatkan akan diolah menjadi sekumpulan data yang siap untuk ditampilkan. Sistem pemilah barang ini didesain agar dapat memilah atau mengelompokkan barang sesuai dengan ketinggian barang dan *barcode* yang ada di barang.

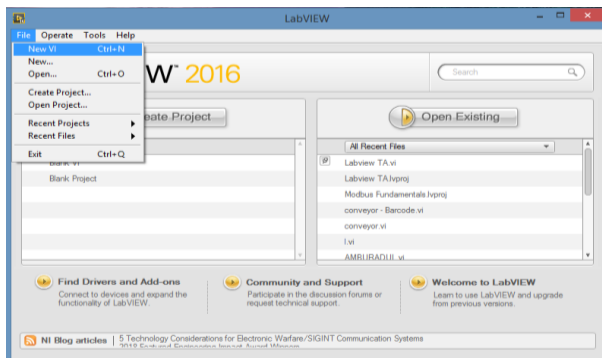
Dalam perancangan penggunaan LabVIEW ini menggunakan *software* LabVIEW dan untuk prosesnya adalah sebagai berikut :

1. Membuka *Software* LabVIEW



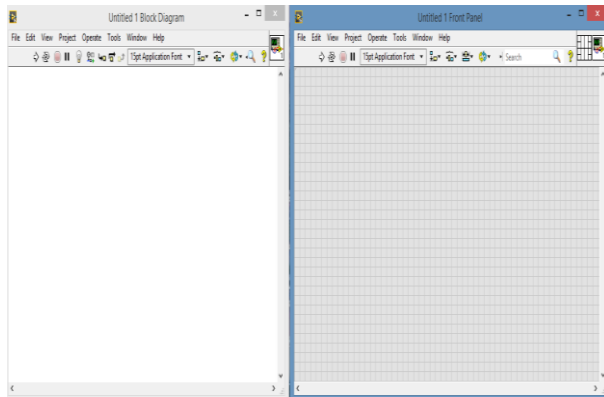
Gambar 3.19 *Software* LabVIEW

2. Pilih *New VI*



Gambar 3.20 Membuka Lembar Kerja Baru

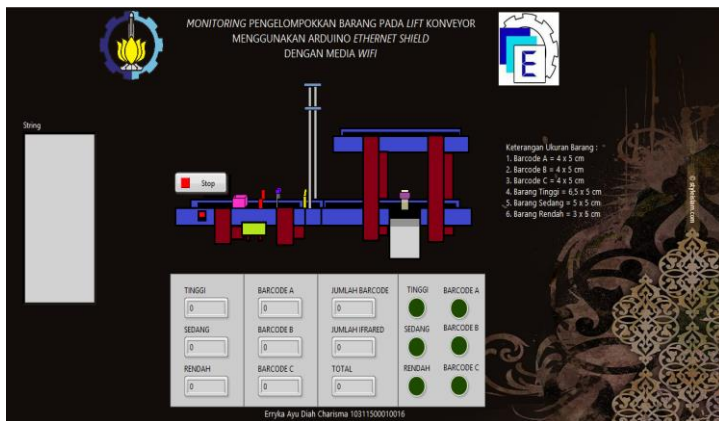
Untuk proses pembuatan lembar kerja baru, yaitu membuka menu *File* kemudian pilih submenu *New VI* dan memunculkan blok diagram dan *front panel* di LabVIEW.



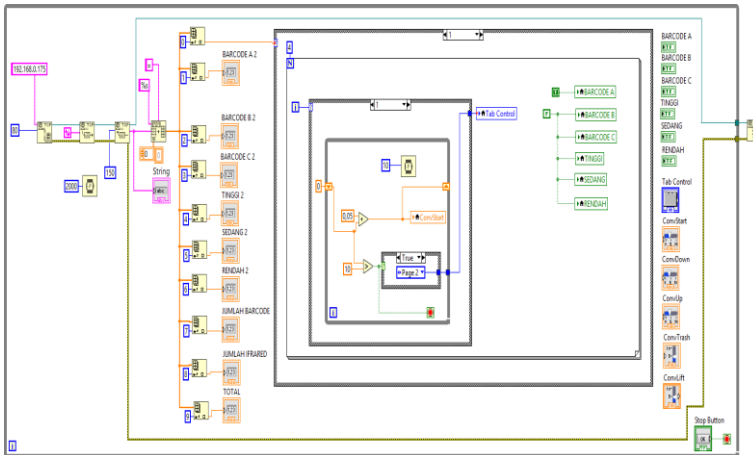
Pada Gambar 3.21 merupakan tampilan lembar kerja LabVIEW yaitu blok diagram dan *front panel* di LabVIEW

3. Membuat Tampilan dan Program pada LabVIEW

Selanjutnya adalah melakukan pemograman pada LabVIEW di blok diagram serta membuat tampilan di *front panel* guna menampilkan hasil data yang diperoleh. Dari Gambar 3.22 merupakan tampilan hasil data dan Gambar 3.23 merupakan proses pemograman yang digunakan.



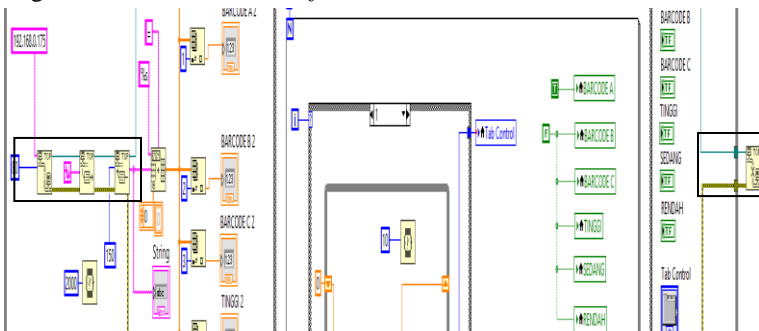
Gambar 3.22 Tampilan pada LabVIEW



Gambar 3.23 Pemograman pada LabVIEW

3.6 Monitoring Sistem Interface

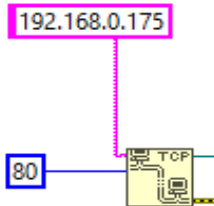
Berikut sistem *interface* data dari Arduino untuk *monitoring* LabVIEW dapat dilihat Gambar 3.24. Dalam proses komunikasi data antar komputer melalui internet atau *wifi* dibutuhkan suatu *protocol* yaitu kumpulan peraturan yang mengatur proses komunikasi antar piranti elektronik salah satunya yaitu *TCP/IP* yang dikenal sebagai *protocol* internet atau *wifi*. *TCP/IP* merupakan *protocol* yang dimanfaatkan sebagai sarana pengiriman data melalui jaringan komputer dengan media internet atau *wifi*.



Gambar 3.24 Blok Diagram Komunikasi Arduino

1. *TCP Open Connection (Transmission Control Protocol)*

Pada program LabVIEW menggunakan *TCP Open Connection (Transmission Control Protocol)* sebagai komunikasi Arduino dan membaca *IP address ethernet* dan mengirim data dari Arduino dengan LabVIEW dapat dilihat pada Gambar 3.25.



Gambar 3.25 *TCP Open Connection*

2. *TCP Write*

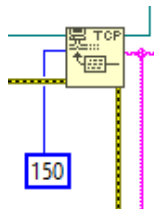
Tahapan ini dilakukan guna menyalurkan data dari *TCP Open Connection* ke *TCP Read* seperti pada Gambar 3.26.



Gambar 3.26 *TCP Write*

3. *TCP Read*

Tahap ini adalah menerima data dari *TCP Write* kemudian disalurkan pada terminal data out pada kotak berwarna biru agar dapat menampilkan hasil data yang diperoleh seperti pada Gambar 3.27.



Gambar 3.27 *TCP Read*

4. TCP Close

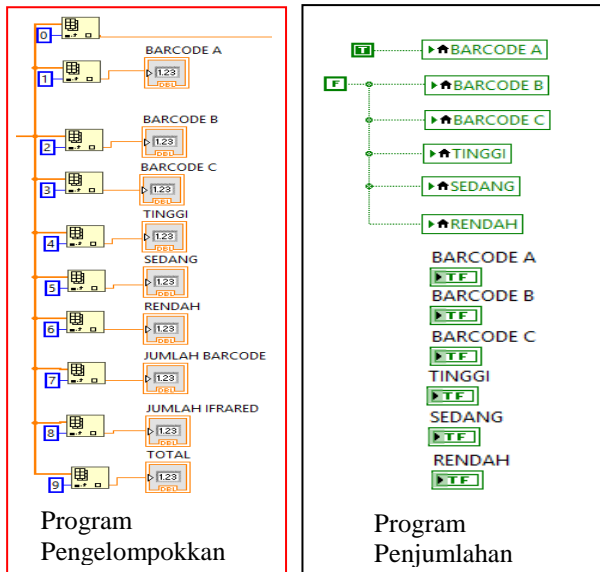
Menghentikan operasi jika adanya kesalahan atau gangguan pada komunikasi serial. Berikut gambat *TCP Close* dapat dilihat pada Gambar 3.28.



Gambar 3.28 *TCP Close*

3.7 Program Pengelompokkan dan Penjumlahan

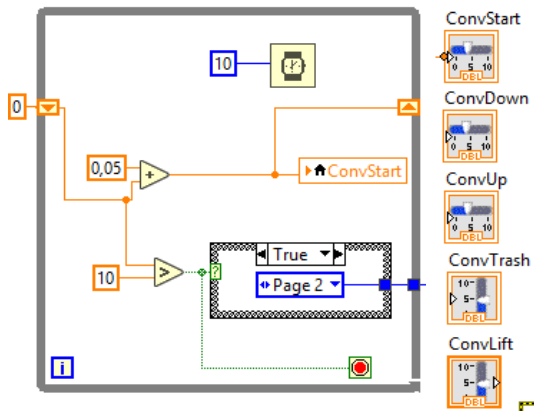
Untuk menampilkan hasil pengelompokkan dan penjumlahan pada LabVIEW dapat dilihat Gambar 3.29. Agar hasil barang yang di deteksi oleh sensor dapat dijumlahkan dan ditampilkan pada *front panel* yaitu dengan membuat program pada blok diagram. Gambar 3.29 pada kotak berwarna merah merupakan program guna menghitung jumlah barang yang dideteksi. Sedangkan dalam pengelompokkan barang menggunakan LED sebagai indikator guna mengetahui barang apa saja yang akan di kelompokkan yaitu dengan memprogram seperti kotak berwarna hitam pada Gambar 3.29.



Gambar 3.29 Program Pengelompokkan dan Penjumlahan

3.8 Program Animasi Konveyor

Tampilan LabVIEW merupakan hal penting guna mengetahui hasil pengelompokkan yang didapatkan. Dari tampilan tersebut juga dapat berupa animasi. Salah satunya adalah barang di konveyor dapat bergerak sesuai apa yang diinginkan dengan suatu program yang ada di blok diagram pada LabVIEW. Untuk lebih jelasnya dapat dilihat pada Gambar 3.30.



Gambar 3. 30 Program Animasi Konveyor

Keterangan :

1. *ConvStart* merupakan proses awal animasi konveyor dimana barang yang terletak pada konveyor 1 pindah ke konveyor 2. Untuk gambarnya dapat dilihat pada Lampiran halaman A-1.
2. *ConvDown* merupakan proses pengelompokkan yang prosesnya dari konveyor 2 menuju ke konveyor 3. Untuk lebih jelasnya gambar dapat dilihat pada Lampiran halaman A-2.
3. *ConvLift* digunakan ketika konveyor 2 berfungsi sebagai *lift* dalam pengelompokkan. Dimana ketika konveyor 1 dalam posisi *ON* barang akan bergerak menuju ke konveyor 2. Kemudian konveyor 2 akan naik dan sejajar dengan konveyor 2. Lebih jelasnya gambar dapat dilihat pada Lampiran halaman A-3
4. *ConvUp* ketika barang sudah berpindah menggunakan *lift* ke konveyor 4 maka secara otomatis konveyor 4 akan *ON* dan

barang akan bergerak menuju ke konveyor 4. Untuk lebih jelasnya gambar dapat dilihat pada Lampiran halaman A-4.

5. *ConvTrash* berfungsi sebagai animasi dari *pusher*. Apabila barang tepat pada *pusher* secara otomatis barang akan di dorong ke tempat pengelompokkan. Untuk lebih jelasnya gambar dapat dilihat pada Lampiran halaman A-5.

BAB IV

PENGUJIAN DAN ANALISA

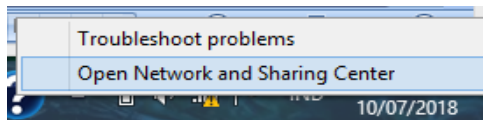
Setelah perancangan dan pembuatan alat selesai, selanjutnya adalah menggabungkan *hardware* dan *software* tersebut sehingga menjadi sebuah sistem kerja. Kemudian melakukan pengujian awal dengan melakukan *monitoring* dan menganalisa data. Berikut *monitoring* dan analisa data yang dilakukan.

4.1 Pengujian Koneksi Wifi dengan PC

Tahap awal yang dilakukan dalam memulai komunikasi antara *ethernet shield* dengan perangkat komputer atau laptop adalah mengkoneksikan *wifi* dengan *PC*. Pengaturan untuk *IP address wifi* dengan cara me-setting *Internet Protocol Version 4* (TCP/IPv4) menjadi otomatis. Langkah-langkah pengaturan *IP address wifi* menjadi otomatis dapat dilihat pada Gambar 4.1, Gambar 4.2, Gambar 4.3, Gambar 4.4, Gambar 4.5 dan Gambar 4.6.

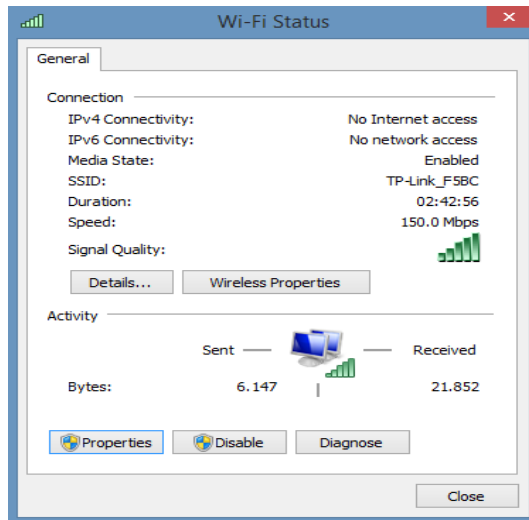
1. Cek koneksi *wifi* terlebih dahulu. Pilih *connect* pada TPLINKF5BC agar dapat terhubung. Untuk mengetahui jika sudah tersambung klik kanan pada koneksi *wifi* dan pilih *Open Network and Sharing Center*.

;

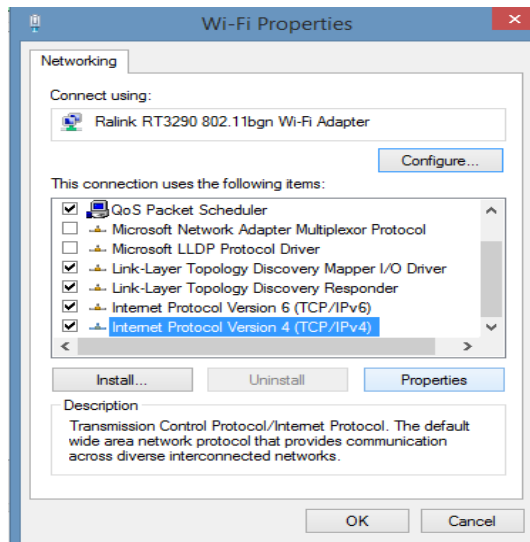


Gambar 4.1 Cara Cek Koneksi Wifi

2. Kemudian muncul seperti Gambar 4.2 dan pilih *properties*. Klik *Internet Protocol Version 4* (TCP/IPv4) seperti Gambar 4.3.

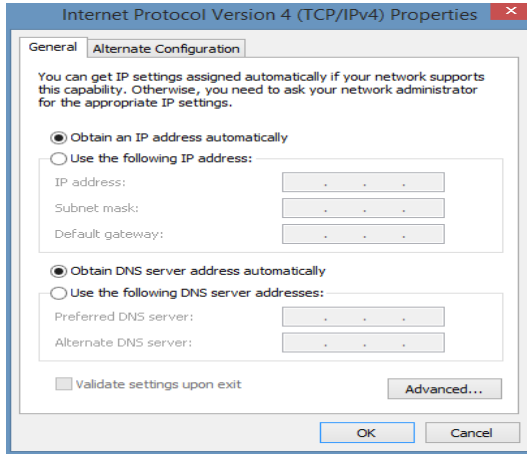


Gambar 4.2 Status Wifi



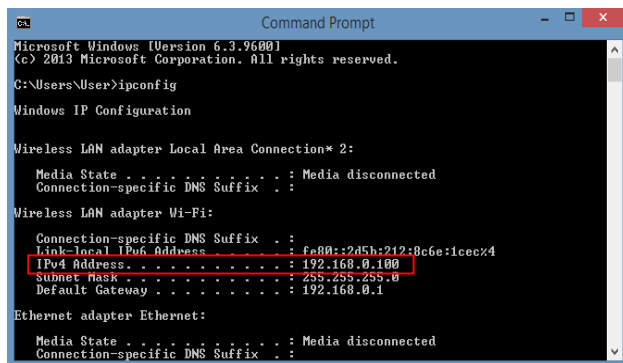
Gambar 4.3 Wifi Properties

3. Setelah pilih *Internet Protocol Version 4 (TCP/IPv4)* kemudian ubah *IP address wifi* menjadi otomatis seperti Gambar 4.4.



Gambar 4.4 Setting IP Address Wifi pada Laptop

4. Disaat melakukan *setting IP address wifi* dan diubah secara otomatis, koneksinya dapat diuji dengan melakukan *ipconfig* pada *command prompt* di laptop. Pengujian koneksi *wifi* pada *command prompt* dapat dilihat pada Gambar 4.5.



Gambar 4.5 Command Prompt dengan IP Address Wifi

Setelah melakukan pengujian koneksi *wifi*, bahwa *wifi* yang tersambung pada laptop dapat digunakan. Disaat mengirim perintah *ipconfig* pada *command prompt* seperti Gambar 4.5 maka laptop memunculkan nomor *IP address* *wifi* sesuai petunjuk gambar diatas dalam kotak berwarna merah yang menandakan jika *wifi* dengan *PC* dapat terhubung.

4.2 Pengujian Koneksi Jarak *Monitoring*

Monitoring merupakan proses pengumpulan data berupa data *real time* dengan beberapa sumber barang dari berbagai jarak. Agar mengetahui koneksi jarak stabil pada *PC* maka perlu melakukan tes ping di *command prompt* terhadap *IP address ethernet shield*.

4.2.1 Pengujian Koneksi Tanpa Penghalang

Pengujian kali ini dilakukan tanpa penghalang, agar mengetahui jarak maksimal yang dicapai dari koneksi *wifi* dan *PC*, maka dilakukan langkah-langkah pengujian sebagai berikut :

1. Pastikan *wifi* dan *PC* telah terhubung, dapat dilihat dari Gambar 4.5.
2. Letakkan *wifi* dan *PC* dengan jarak hingga 150 meter secara berurutan tanpa penghalang.
3. Lakukan tes ping *IP address* pada setiap jarak di *command prompt*.

Dalam menunjukkan jarak antara *PC* dan alat dapat dilihat dari *Google Maps* dengan Gambar 4.6. Sedangkan pengujian jarak pada *command prompt* dapat dilihat Gambar 4.7, Gambar 4.8, Gambar 4.9, Gambar 4.10, dan Gambar 4.11.



Gambar 4.6 Jarak *PC* dan *Alat* dari *Google Maps*

Posisi untuk alat berada di titik merah seperti Gambar 4.6 dan *PC* berada di jarak maksimal 150 meter. Dalam *monitoring* perlu mengetahui jarak antara alat dan *PC* agar dapat mengetahui seberapa lancar *me-monitoring*. Untuk pengambilan data dilakukan dengan cara barang ditempatkan di konveyor 1 yang sedang berjalan kemudian barang akan di deteksi oleh salah satu sensor. Hasil data yang diperoleh dari Arduino dikirim ke LabVIEW untuk ditampilkan. Sedangkan *PC* diletakkan jauh dengan alat yang berjarak hingga 150 meter. Dari jarak tersebut dilakukan tanpa ada halangan apapun dan sedang berada di luar ruangan.

```
C:\Users\User>ping 192.168.0.175

Pinging 192.168.0.175 with 32 bytes of data:
Reply from 192.168.0.175: bytes=32 time=3ms TTL=128
Reply from 192.168.0.175: bytes=32 time=4ms TTL=128
Reply from 192.168.0.175: bytes=32 time=2ms TTL=128
Reply from 192.168.0.175: bytes=32 time=2ms TTL=128

Ping statistics for 192.168.0.175:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 4ms, Average = 2ms
```

Gambar 4.7 Cek *Loss* Koneksi Jarak 20 Meter

```

C:\Users\User>ping 192.168.0.175

Pinging 192.168.0.175 with 32 bytes of data:
Reply from 192.168.0.175: bytes=32 time=8ms TTL=128
Reply from 192.168.0.175: bytes=32 time=4ms TTL=128
Reply from 192.168.0.175: bytes=32 time=7ms TTL=128
Reply from 192.168.0.175: bytes=32 time=15ms TTL=128

Ping statistics for 192.168.0.175:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 4ms, Maximum = 15ms, Average = 8ms

```

Gambar 4.8 Cek *Loss* Koneksi Jarak 50 Meter

```

C:\Users\User>ping 192.168.0.175

Pinging 192.168.0.175 with 32 bytes of data:
Reply from 192.168.0.175: bytes=32 time=162ms TTL=128
Reply from 192.168.0.175: bytes=32 time=1ms TTL=128
Reply from 192.168.0.175: bytes=32 time=36ms TTL=128
Reply from 192.168.0.175: bytes=32 time=1ms TTL=128

Ping statistics for 192.168.0.175:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 162ms, Average = 50ms

```

Gambar 4.9 Cek *Loss* Koneksi Jarak 120 Meter

Dapat dilihat dari Gambar 4.7, Gambar 4.8, dan Gambar 4.9 menunjukkan bahwa *wifi* dapat terhubung laptop dengan jarak hingga 120 meter tanpa adanya *loss* koneksi yang terdeteksi sehingga koneksi berjalan lancar tanpa adanya gangguan. Dibuktikan ketika mengirim perintah ping pada *command prompt*.

```

C:\Users\User>ping 192.168.0.175

Pinging 192.168.0.175 with 32 bytes of data:
Request timed out.
Request timed out.
Reply from 192.168.0.175: bytes=32 time=174ms TTL=128
Request timed out.

Ping statistics for 192.168.0.175:
    Packets: Sent = 4, Received = 1, Lost = 3 (75% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 174ms, Maximum = 174ms, Average = 174ms

```

Gambar 4.10 Cek *Loss* Koneksi Jarak 130 Meter

Sedangkan Gambar 4.10 merupakan untuk tes ping pada *command prompt* dan terdapat 75% *loss* koneksi dengan jarak 130 meter. Sehingga koneksi sering mengalami gangguan yang

mengakibatkan *monitoring* sulit dalam menampilkan hasil data yang diterima. Gangguan yang dimaksud ialah koneksi *wifi* sering terputus.

```
C:\Users\User>ping 192.168.0.175

Pinging 192.168.0.175 with 32 bytes of data:
Request timed out.
Request timed out.
General failure.
General failure.

Ping statistics for 192.168.0.175:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Gambar 4.11 Cek *Loss* Koneksi Jarak 150 Meter

Ketika 100% *loss* maka koneksi *wifi* terputus dengan jarak 150 meter. Sehingga sensor pada konveyor tidak dapat mendeteksi barang, meskipun terdapat barang yang lewat pada konveyor dan tidak dapat *me-monitoring* pengelompokkan barang. Hal ini dapat dibuktikan dengan tes ping pada *command prompt* seperti Gambar 4.11.

Tabel 4.1 Hasil Pengujian Koneksi Tanpa Penghalang

Jarak (meter)	Loss (%)	Time Average
10 meter	0 %	2 ms
20 meter	0 %	8 ms
30 meter	0 %	13 ms
40 meter	0 %	15 ms
50 meter	0 %	20 ms
60 meter	0 %	25 ms
70 meter	0 %	25 ms
80 meter	0 %	28 ms
90 meter	0 %	34 ms
100 meter	0 %	45ms
110 meter	0 %	42 ms
120 meter	0 %	50 ms
130 meter	75 %	174 ms
140 meter	75 %	510 ms
150 meter	100 %	Time Out

Pada Tabel 4.1 merupakan hasil pengujian koneksi komunikasi tanpa penghalang. Dari setiap jarak memiliki perbedaan dalam koneksi *loss* maupun *time average*. Koneksi *loss* merupakan presentase dalam

pengukuran jarak komunikasi yang diperoleh, sedangkan *time average* rata-rata waktu ketika pengiriman data ke LabVIEW untuk ditampilkan.

Dalam pengujian jarak kali ini setiap proses pada tes ping *command prompt* dilakukan dengan uji jarak per 10 meter. Untuk hasil dari tes ping *command prompt* dapat dilihat di Gambar 4.7, Gambar 4.8, Gambar 4.9, Gambar 4.10, dan Gambar 4.11 dan lebih jelasnya dapat dilihat pada Tabel 4.1.

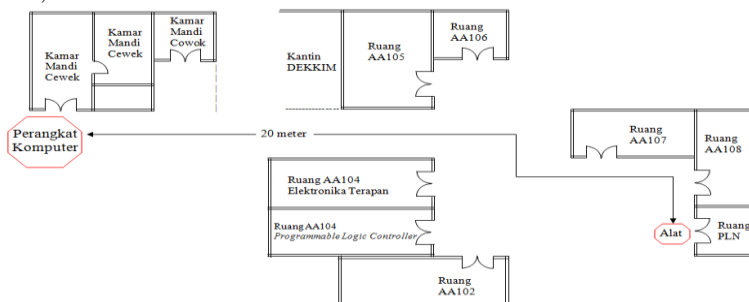
Sehingga dapat disimpulkan bahwa pengujian jarak *monitoring* pengelompokkan barang dapat dilakukan dengan jarak hingga 120 meter. Dikarenakan pada jarak 130 meter hingga 150 meter mengalami gangguan berupa koneksi pada *wifi* yang sering terputus maupun terputus.

4.2.2 Pengujian Koneksi Dengan Penghalang

Pengujian kali ini dilakukan dengan penghalang dari gedung dan pohon-pohonan. Agar mengetahui jarak maksimal yang dicapai dari koneksi *wifi* dan *PC*, maka dilakukan langkah-langkah pengujian sebagai berikut :

1. Pastikan *wifi* dan *PC* telah terhubung, dapat dilihat dari Gambar 4.7.
2. Letakkan *wifi* dan *PC* dengan jarak hingga 20 meter secara berurutan dengan penghalang.
3. Lakukan tes ping *IP address* pada setiap jarak di *command prompt*.

Dalam menunjukkan jarak antara *PC* dan *wifi* dapat dilihat dari denah dengan Gambar 4.12. Sedangkan gambar pengujian jarak pada *command prompt* dapat dilihat Gambar 4.13, Gambar 4.14, Gambar 4.15, dan Gambar 4.16.



Gambar 4.12 Denah Jarak *PC* dan Alat

Posisi untuk alat dan *PC* berada pada jarak hingga 20 meter. Sama halnya dengan pengujian koneksi tanpa penghalang bahwa disaat me- *monitoring* perlu mengetahui jarak antara alat dan *monitoring* agar dapat me-*monitoring* sesuai jarak yang ditentukan. Untuk pengambilan data dilakukan dengan cara barang ditempatkan di konveyor yang sedang berjalan kemudian barang di deteksi oleh salah satu sensor yang aktif. Hasil data yang diperoleh dari Arduino dikirim ke LabVIEW untuk ditampilkan melalui *wifi* yang terhubung. Dalam pengujian koneksi terlebih dahulu menentukan jarak yaitu 5 meter, 10 meter, 15 meter, dan 20 meter. Dari jarak tersebut, *monitoring* diletakkan jauh dengan *wifi* serta konveyor dengan terdapat halangan berupa gedung.

```
C:\Users\User>ping 192.168.0.175

Pinging 192.168.0.175 with 32 bytes of data:
Reply from 192.168.0.175: bytes=32 time=1ms TTL=128
Reply from 192.168.0.175: bytes=32 time=5ms TTL=128
Reply from 192.168.0.175: bytes=32 time=3ms TTL=128
Reply from 192.168.0.175: bytes=32 time=4ms TTL=128

Ping statistics for 192.168.0.175:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 5ms, Average = 3ms
```

Gambar 4.13 Cek *Loss* Koneksi Jarak 5 Meter

```
C:\Users\User>ping 192.168.0.175

Pinging 192.168.0.175 with 32 bytes of data:
Reply from 192.168.0.175: bytes=32 time=1ms TTL=128
Reply from 192.168.0.175: bytes=32 time=4ms TTL=128
Reply from 192.168.0.175: bytes=32 time=1ms TTL=128
Reply from 192.168.0.175: bytes=32 time=3ms TTL=128

Ping statistics for 192.168.0.175:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 4ms, Average = 2ms
```

Gambar 4.14 Cek *Loss* Koneksi Jarak 10 Meter

```
C:\Users\User>ping 192.168.0.175

Pinging 192.168.0.175 with 32 bytes of data:
Reply from 192.168.0.175: bytes=32 time=312ms TTL=128
Reply from 192.168.0.175: bytes=32 time=470ms TTL=128
Reply from 192.168.0.175: bytes=32 time=458ms TTL=128
Reply from 192.168.0.175: bytes=32 time=801ms TTL=128

Ping statistics for 192.168.0.175:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 312ms, Maximum = 801ms, Average = 510ms
```

Gambar 4.15 Cek *Loss* Koneksi Jarak 15 Meter

Dapat dilihat dari Gambar 4.13, Gambar 4.14, dan Gambar 4.15 menunjukkan bahwa *wifi* dapat terhubung laptop dengan jarak sejauh 5 meter, 10 meter, dan 15 meter tidak terdapat adanya *loss* koneksi yang terdeteksi sehingga koneksi berjalan lancar tanpa ada gangguan. Dibuktikan ketika mengirim perintah ping pada *command prompt*.

```
C:\Users\User>ping 192.168.0.175

Pinging 192.168.0.175 with 32 bytes of data:
Reply from 192.168.0.175: bytes=32 time=83ms TTL=128
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.0.175:
    Packets: Sent = 4, Received = 1, Lost = 3 (75% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 83ms, Maximum = 83ms, Average = 83ms
```

Gambar 4.16 Cek *Loss* Koneksi Jarak 20 Meter

Sedangkan Gambar 4.16 merupakan untuk tes ping pada *command prompt* dan terdapat 75% *loss* koneksi dengan jarak 20 meter. Sehingga koneksi akan terputus yang mengakibatkan *monitoring* tidak dapat menampilkan hasil data meskipun terdapat barang yang lewat pada konveyor.

Tabel 4.2 Hasil Pengujian Koneksi Dengan Penghalang

Jarak (meter)	Loss (%)	Time Average
5 meter	0 %	3 ms
10 meter	0 %	2 ms
15 meter	0 %	510 ms
20 meter	75 %	83 ms

Pada Tabel 4.2 merupakan hasil pengujian koneksi komunikasi dengan penghalang. Dari setiap jarak memiliki perbedaan koneksi *loss* maupun *time average*. Koneksi *loss* merupakan presentase dalam pengukuran jarak komunikasi yang dapat di *monitoring*, sedangkan *time average* rata-rata waktu ketika pengiriman hasil data ke LabVIEW agar dapat ditampilkan.

Dalam pengujian jarak kali ini setiap proses pada tes ping *command prompt* dilakukan dengan uji jarak per 5 meter. Untuk hasil dari tes ping *command prompt* dapat dilihat di Gambar 4.13, Gambar 4.14, Gambar 4.15 dan Gambar 4.16 dan lebih jelasnya dapat dilihat pada Tabel 4.2.

Sehingga dapat disimpulkan bahwa pengujian jarak *monitoring* pengelompokkan barang dapat dilakukan dengan jarak hingga 15 meter dengan halangan gedung. Dikarenakan pada jarak 20 meter mengalami gangguan berupa koneksi pada *wifi* terputus.

4.3 Pengujian Keseluruhan Sistem Pada Software LabVIEW

Setelah dilakukan pengujian koneksi *wifi* dengan *PC* tanpa penghalang dan dengan penghalang. Kemudian melakukan pengujian sensor *infrared* dan *scanner barcode* pada *software* LabVIEW yang bertujuan untuk mengetahui kinerja keseluruhan dari alat ini. Pada gambar di bawah ini merupakan tampilan awal LabVIEW dari pengelompokkan barang. Tampilan awal ini tidak mendapatkan hasil data yang diperoleh dari Arduino dengan mendeteksi barang oleh sensor *infrared* dan *scanner barcode* di konveyor. Sehingga pada tampilan LED tidak menyala, barang pada konveyor tidak berpindah tempat, dan penghitungan berjumlah 0.



Gambar 4.17 Tampilan Awal LabVIEW

4.3.1 Pengujian Sensor *Infrared*

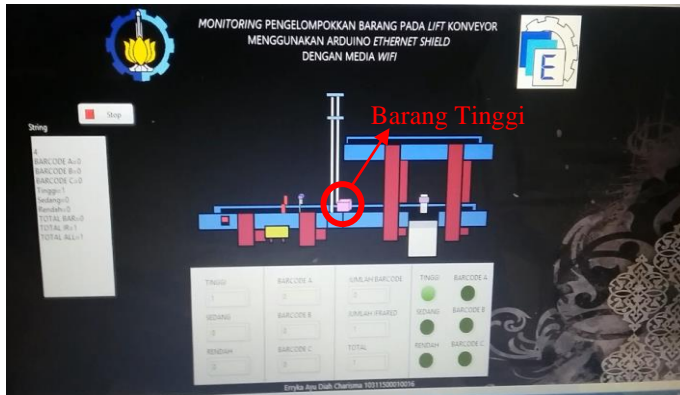
Pada pengujian kali ini dilakukan dengan mendeteksi barang sesuai ketinggian oleh sensor *infrared*. Pendeteksian barang dilakukan dengan tiga bentuk barang yaitu barang tinggi, barang sedang, dan barang rendah. Pengujian ini menggunakan sensor *infrared*, *relay*, konveyor, Arduino, *ethernet shield*, dan *wifi*.

Langkah-langkah pengujian ini yaitu dengan cara merangkai *relay* dengan Arduino seperti Gambar 3.4, merangkai *power supply*

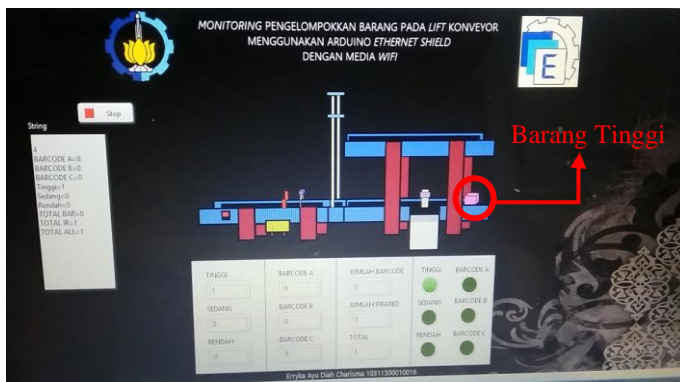
dengan *relay* seperti Gambar 3.5, merangkai *relay* dengan motor seperti Gambar 3.6, dan merangkai sensor *infrared*, Arduino, *ethernet shield*, serta *wifi* seperti Gambar 3.12. Kemudian, *upload* program Arduino yang dapat dilihat pada Lampiran B halaman B-1 hingga halaman B-15. Aktifkan *push button*, maka konveyor 1 akan berjalan. Kemudian, sambungkan Arduino dengan sumber, maka sensor *infrared* akan aktif. Selanjutnya, letakkan barang tinggi di konveyor maka barang akan berjalan mendekati sensor *infrared* kemudian mengelompokkannya sesuai yang telah ditentukan. Pada awalnya barang akan diletakkan pada konveyor 1, ketika barang sudah berada di konveyor 1 maka barang tersebut akan dideteksi oleh sensor *infrared* yang tersusun secara *vertical* dalam sebuah *panel box* dan menuju ke konveyor 3 untuk lebih jelasnya dapat dilihat seperti gambar pada Lampiran A.6 halaman A-6. Untuk tampilan di LabVIEW barang tinggi juga berjalan di konveyor 1 dan menuju ke konveyor 3 seperti Gambar 4.18, Gambar 4.19, dan Gambar 4.20. Setelah proses pengelompokan selesai, maka LabVIEW akan menampilkan hasil perhitungan pada Gambar 4.21. Kemudian, barang dapat di *input* kan kembali setelah proses pengelompokan selesai. Setelah barang tinggi di *input* kan sebanyak 10 kali untuk pengujian, maka tekan tombol *reset* pada Arduino dan ambil data pengujian untuk barang lainnya yang berdasarkan ketinggian sebanyak 10 kali. Dan berikut ini adalah gambar *step-step* dari pengelompokan barang tinggi pada LabVIEW.



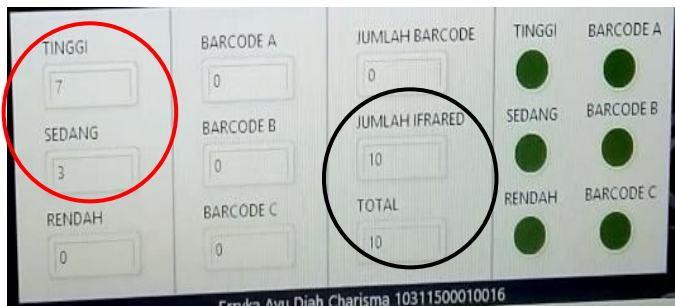
Gambar 4.18 Barang Tinggi Bergerak di Konveyor 1



Gambar 4.19 Barang Tinggi Melewati Konveyor 2



Gambar 4.20 Kondisi Ketika Barang Tinggi Dikelompokkan



Gambar 4.21 Hasil Penjumlahan Barang Tinggi

Ketika sensor *infrared* di konveyor mendeteksi barang tinggi maka barang tersebut akan berjalan dari konveyor 1 hingga konveyor 3, proses tersebut dapat dilihat pada Lampiran A.6 halaman A-6. Begitu pula pada tampilan LabVIEW, untuk Gambar 4.18 barang berjalan di konveyor 1 dan LED menyala di kondisi barang tinggi. Kemudian barang berjalan melewati konveyor 2 seperti pada Gambar 4.19. Barang masih berjalan ke konveyor 3 hingga sampai di ujung konveyor yang ditunjukkan pada Gambar 4.20. Setelah pengelompokkan barang tinggi pada LabVIEW telah selesai maka LED yang awalnya menyala di kondisi barang tinggi akan mati dan menampilkan hasil penjumlahan barang tinggi yang di deteksi. Untuk lingkaran berwarna merah merupakan hasil jumlah barang yang terdeteksi oleh sensor. Selain itu, lingkaran berwarna hitam jumlah keseluruhan hasil deteksi barang pada sensor *infrared* seperti Gambar 4.21. Hasil penjumlahan tersebut di dapatkan dari pengujian deteksi barang tinggi dengan 10 kali percobaan yang dilakukan secara berurutan untuk lebih jelasnya dapat dilihat pada Tabel 4.3.

Tabel 4.3 Hasil Pengujian Barang Tinggi

Barang Tinggi Ke-	Hasil Tampilan <i>Monitoring</i>	Keterangan
1	Barang Tinggi	Sesuai
2	Barang Tinggi	Sesuai
3	Barang Tinggi	Sesuai
4	Barang Sedang	Tidak Sesuai
5	Barang Tinggi	Sesuai
6	Barang Tinggi	Sesuai
7	Barang Sedang	Tidak Sesuai
8	Barang Tinggi	Sesuai
9	Barang Sedang	Tidak Sesuai
10	Barang Tinggi	Sesuai

Dari Tabel 4.3 dapat dilihat ketika melakukan 10 kali percobaan terdapat kesalahan sebanyak 3 kali yang ditunjukkan dengan ketidaksesuaian dalam *monitoring*. Untuk Tabel 4.3 pengujian dikatakan tidak sesuai apabila hasil *monitoring* pada LabVIEW mendeteksi barang tinggi. Sedangkan pengujian dikatakan sesuai jika hasil *monitoring* pada LabVIEW mendeteksi barang tinggi. Dengan begitu, presentase *error* pada pengujian keseluruhan dari barang tinggi sebesar 70%. Hasil presentase *error* yang diperoleh di dapatkan dari Persamaan 4.1.

$$\%Error = \frac{\text{Jumlah Error}}{\text{Jumlah Pengujian}} \times 100\% \dots\dots\dots(4.1)$$

$$\%Error = \frac{3}{10} \times 100\%$$

$$\%Error = 70\%$$

Dari 10 percobaan tersebut, hanya 7 percobaan yang berhasil mendeteksi dengan benar. Kesalahan yang didapatkan karena meletakkan barang tidak tepat ataupun pencahayaan yang terlalu terang. Setelah pengujian barang tinggi selesai, maka tekan tombol *reset* pada Arduino. Selanjutnya, *input* barang sedang di konveyor maka barang akan berjalan mendekati sensor *infrared*. Ketika barang sedang berjalan di konveyor 1 dan menuju ke konveyor 4 terlebih dahulu harus melewati konveyor 2 yang digunakan sebagai *lift* guna mengelompokkan barang untuk lebih jelasnya dapat dilihat pada Lampiran A.7 halaman A-7. Begitu pun untuk tampilan di LabVIEW, *barcode* A berjalan di konveyor 1 untuk menuju ke konveyor 4 terlebih dahulu harus melewati konveyor 2 yang digunakan sebagai *lift* seperti Gambar 4.22, Gambar 4.23, dan Gambar 4.24. Setelah proses pengelompokan selesai, maka LabVIEW akan menampilkan hasil perhitungan pada Gambar 4.25. Kemudian, barang dapat di *input* kan kembali setelah proses pengelompokan selesai. Setelah barang sedang di *input* kan sebanyak 10 kali untuk pengujian, maka tekan tombol *reset* pada Arduino dan ambil data pengujian untuk barang lainnya berdasarkan ketinggian sebanyak 10 kali. Dan berikut ini adalah gambar *step- step* dari pengelompokan barang sedang pada LabVIEW.



Gambar 4.22 Barang Sedang Bergerak di Konveyor 1



Gambar 4.23 Barang Sedang Naik *Lift* Melalui Konveyor 2



Gambar 4.24 Kondisi Ketika Barang Sedang Dikelompokkan



Gambar 4.25 Hasil Penjumlahan Sedang

Ketika sensor *infrared* di konveyor mendeteksi barang sedang maka barang tersebut akan berjalan dari konveyor 1 hingga konveyor 4 melalui konveyor 2 yang sebagai *lift*, proses tersebut dapat dilihat pada Lampiran A.7 halaman A-7. Begitu pula pada tampilan LabVIEW, untuk Gambar 4.22, barang berjalan di konveyor 1 dan LED menyala di kondisi barang sedang. Kemudian barang naik *lift* melalui konveyor 2 seperti pada Gambar 4.23. Disaat konveyor 2 sejajar dengan konveyor 4 maka barang berjalan ke konveyor 4 hingga sampai di ujung konveyor yang ditunjukkan pada Gambar 4.24. Setelah pengelompokkan barang sedang pada LabVIEW telah selesai maka LED yang awalnya menyala di kondisi barang sedang akan mati dan menampilkan hasil penjumlahan barang sedang yang di deteksi. Untuk lingkaran berwarna merah merupakan hasil jumlah barang yang terdeteksi oleh sensor. Selain itu, lingkaran berwarna hitam jumlah keseluruhan hasil deteksi barang pada sensor *infrared* seperti Gambar 4.25. Hasil penjumlahan tersebut di dapatkan dari pengujian deteksi barang sedang dengan 10 kali percobaan yang dilakukan secara berurutan untuk lebih jelasnya dapat dilihat pada Tabel 4.4.

Tabel 4.4 Hasil Pengujian Barang Sedang

Barang Sedang Ke-	Hasil Tampilan <i>Monitoring</i>	Keterangan
1	Barang Sedang	Sesuai
2	Barang Rendah	Tidak Sesuai
3	Barang Sedang	Sesuai
4	Barang Sedang	Sesuai
5	Barang Sedang	Sesuai
6	Barang Rendah	Tidak Sesuai
7	Barang Sedang	Sesuai
8	Barang Sedang	Sesuai
9	Barang Sedang	Sesuai
10	Barang Sedang	Sesuai

Berdasarkan Tabel 4.4 didapatkan bahwa, ketika barang sedang terdeteksi di konveyor 1 maka kondisi ketika *relay* 2 mengaktifkan motor konveyor 2 kemudian *relay* 5 aktif untuk mengaktifkan motor pada *lift* agar dapat berjalan keatas atau naik, lalu *relay* 2 mengaktifkan motor konveyor 2 dan *relay* 4 untuk mengaktifkan motor konveyor 4. Ketika barang sudah dikelompokkan, secara otomatis *relay* 6 aktif untuk mengaktifkan motor yang menurunkan *lift* agar kembali ke semula.

Dari Tabel 4.4 dapat dilihat ketika melakukan 10 kali percobaan terdapat kesalahan sebanyak 2 kali dengan presentase *error* sebesar 80%. Hasil presentase diperoleh dari kutipan Persamaan 4.1. Untuk Tabel 4.4 pengujian dikatakan tidak sesuai apabila hasil *monitoring* pada LabVIEW mendeteksi barang sedang Sedangkan pengujian dikatakan sesuai jika hasil *monitoring* pada LabVIEW mendeteksi barang sedang. Dari 10 percobaan tersebut, hanya 8 percobaan yang berhasil mendeteksi dengan benar karena meletakkan barang tidak tepat ataupun pencahayaan yang terlalu terang.

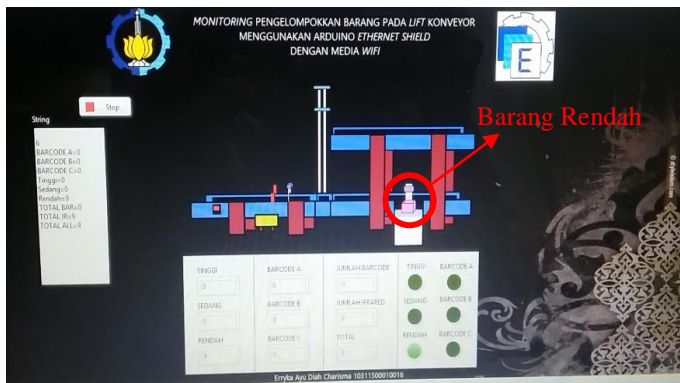
Setelah pengujian barang sedang selesai, maka tekan tombol *reset* pada Arduino. Selanjutnya, *input* barang rendah di konveyor maka barang akan berjalan mendekati sensor *infrared*. Ketika barang rendah berjalan di konveyor 1 dan menuju ke *pusher* yang terletak pada tengah-tengah konveyor 3 untuk lebih jelasnya dapat dilihat pada Lampiran A.8 halaman A-9. Begitu pun untuk tampilan di LabVIEW, barang rendah berjalan di konveyor 1 dan menuju ke *pusher* seperti Gambar 4.26, Gambar 4.27, Gambar 4.28 dan Gambar 4.29. Setelah proses pengelompokan selesai, maka LabVIEW akan menampilkan hasil perhitungan pada Gambar 4.30. Kemudian, barang dapat di *input* kan kembali setelah proses pengelompokan selesai. Setelah barang rendah di *input* kan sebanyak 10 kali untuk pengujian kemudian ambil data pengujian. Dan berikut ini adalah gambar *step-step* dari pengelompokan barang rendah pada LabVIEW.



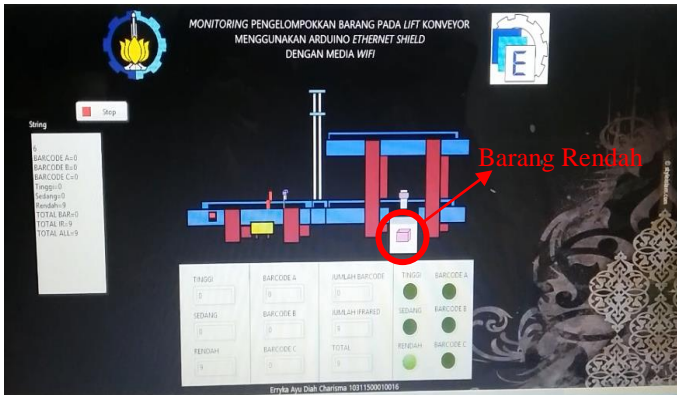
Gambar 4.26 Barang Rendah Bergerak di Konveyor 1



Gambar 4.27 Barang Rendah Melewati Konveyor 2



Gambar 4.28 Pusher Mendorong Barang Rendah



Gambar 4.29 Kondisi Ketika Barang Rendah Dikelompokkan



Gambar 4.30 Hasil Penjumlahan Rendah

Ketika sensor *infrared* di konveyor mendeteksi barang rendah maka barang tersebut akan berjalan dari konveyor 1 ke *pusher*, proses tersebut dapat dilihat pada Lampiran A.8 halaman A-9 . Begitu pula pada tampilan LabVIEW, untuk Gambar 4.26, barang berjalan di konveyor 1 dan LED menyala di kondisi barang rendah. Kemudian barang berjalan melewati konveyor 2 seperti pada Gambar 4.27. Barang masih berjalan hingga menuju *pusher* dan barang tersebut di dorong oleh *pusher* yang ditunjukkan pada Gambar 4.28. Setelah pengelompokkan barang rendah pada LabVIEW telah selesai maka LED yang awalnya menyala di kondisi barang rendah akan mati dan menampilkan hasil penjumlahan barang rendah yang di deteksi. Untuk lingkaran berwarna merah merupakan hasil jumlah barang yang terdeteksi oleh sensor. Selain itu, lingkaran berwarna hitam jumlah keseluruhan hasil deteksi

barang pada sensor *infrared* seperti Gambar 4.30. Hasil penjumlahan tersebut di dapatkan dari pengujian deteksi barang rendah dengan 10 kali percobaan yang dilakukan secara berurutan untuk lebih jelasnya dapat dilihat pada Tabel 4.5.

Tabel 4.5 Hasil Pengujian Barang Rendah

Barang Rendah Ke-	Hasil Tampilan <i>Monitoring</i>	Keterangan
1	Barang Rendah	Sesuai
2	Barang Rendah	Sesuai
3	Barang Rendah	Sesuai
4	Barang Rendah	Sesuai
5	Barang Rendah	Sesuai
6	Barang Rendah	Sesuai
7	Barang Rendah	Sesuai
8	Barang Rendah	Sesuai
9	Barang Rendah	Sesuai
10	Barang Rendah	Sesuai

Berdasarkan Tabel 4.8 didapatkan bahwa, ketika barang rendah terdeteksi di konveyor 1 maka kondisi *relay* 2 mengaktifkan konveyor 2 dan *relay* 3 aktif untuk mengaktifkan konveyor 3 kemudian *relay* 7 aktif ketika barang telah sampai di depan *pusher*, kemudian *pusher* akan mendorongnya. Setelah selesai, *relay* 8 akan aktif untuk mengembalikan *pusher* dalam keadaan semula. Dari Tabel 4.5 dapat diketahui ketika melakukan 10 kali percobaan tidak terdapat kesalahan berupa *monitoring* maupun pendeteksian barang rendah. Untuk Tabel 4.3 pengujian dikatakan sesuai jika hasil *monitoring* pada LabVIEW mendeteksi barang rendah.

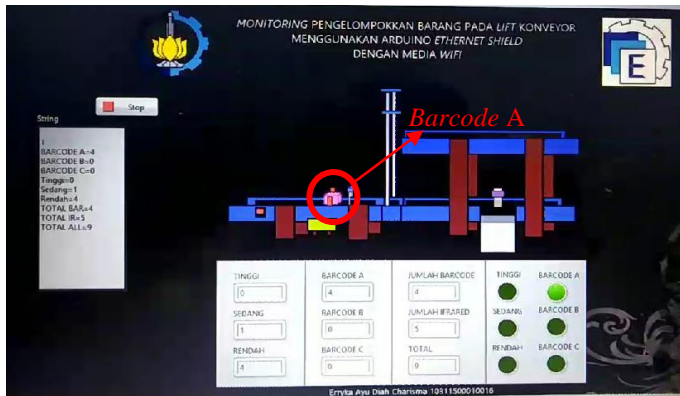
Dari total pengujian dilakukan sebanyak 30 kali dan terdapat kesalahan sebanyak 5 kali. Untuk presentase *error* diperoleh sebesar 16% yang dikutip pada Persamaan 4.1. Kesalahan tersebut berupa menempatkan barang yang tidak sesuai dan pencahayaan terlalu terang.

4.3.2 Pengujian Scanner Barcode

Setelah dilakukan pengujian koneksi *wifi* dengan *PC* tanpa penghalang dan dengan penghalang. Kemudian melakukan pengujian *scanner barcode* di konveyor yang bertujuan untuk mengetahui kinerja keseluruhan dari alat ini. Pengujian dilakukan menggunakan 3 contoh

barang berlabel *barcode* yang dapat dilihat pada Lampiran A.9 halaman A-12. Pengujian ini menggunakan *scanner barcode*, *relay*, konveyor, Arduino, *ethernet shield*, dan *wifi*.

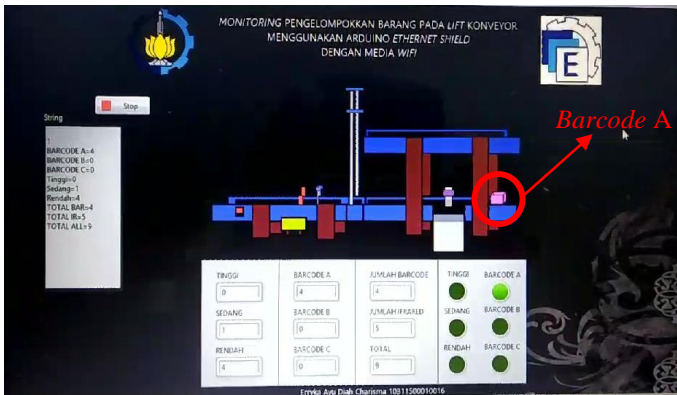
Langkah-langkah pengujian ini yaitu dengan cara merangkai *relay* dengan Arduino seperti Gambar 3.4, merangkai *power supply* dengan *relay* seperti Gambar 3.5, merangkai *relay* dengan motor seperti Gambar 3.6, dan merangkai *scanner barcode*, Arduino, *ethernet shield*, serta *wifi* seperti Gambar 3.12. Kemudian, *upload* program Arduino yang dapat dilihat pada Lampiran B halaman B-1 hingga halaman B-15. Aktifkan *push button*, maka konveyor 1 akan berjalan. Kemudian, sambungkan Arduino dengan sumber, maka *scanner barcode* akan aktif. Selanjutnya, *input* barang berlabel *barcode* A di konveyor maka barang akan berjalan mendekati *scanner barcode* hingga berbunyi “tit” yang menandakan *barcode* terdeteksi kemudian mengelompokkannya sesuai yang telah ditentukan. Ketika *barcode* A berjalan di konveyor 1 yang akan menuju ke konveyor 3 guna mengelompokkan barang dapat dilihat seperti gambar pada Lampiran A.10 halaman A-13 maka tampilan di LabVIEW *barcode* A juga berjalan di konveyor 1 dan menuju ke konveyor 3 seperti Gambar 3.31, Gambar 3.32, dan Gambar 3.33. Setelah proses pengelompokan selesai, maka LabVIEW akan menampilkan hasil perhitungan pada Gambar 3.34. Kemudian, barang dapat di *input* kan kembali setelah proses pengelompokan selesai. Setelah *barcode* yang sama di *input* kan sebanyak 10 kali untuk pengujian, maka tekan tombol *reset* pada Arduino dan ambil data pengujian untuk *barcode* lainnya sebanyak 10 kali. Dan berikut ini adalah gambar *step-step* dari pengelompokan barang *barcode* A pada LabVIEW.



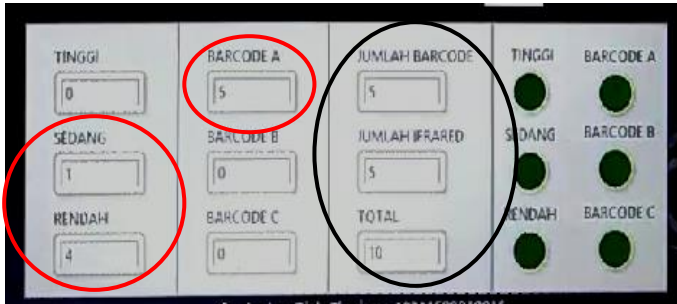
Gambar 4.31 Barang *Barcode A* Bergerak di Konveyor 1



Gambar 4.32 Barang *Barcode A* Melewati Konveyor 2



Gambar 4.33 Kondisi Ketika Barang *Barcode A* Dikelompokkan



Gambar 4.34 Hasil Penjumlahan *Barcode A*

Ketika *scanner barcode* di konveyor mendeteksi *barcode A* maka barang tersebut akan berjalan dari konveyor 1 hingga konveyor 3, proses tersebut dapat dilihat pada Lampiran A.10 halaman A-13 . Begitu pula pada tampilan LabVIEW, untuk Gambar 4.31 barang berjalan di konveyor 1 dan LED menyala di kondisi *barcode A*. Kemudian barang berjalan melewati konveyor 2 seperti pada Gambar 4.32. Barang masih berjalan hingga melewati konveyor 3 hingga sampai di ujung konveyor yang ditunjukkan pada Gambar 4.33. Setelah pengelompokkan *barcode A* pada LabVIEW telah selesai maka LED yang awalnya menyala di kondisi *barcode A* akan mati dan menampilkan hasil penjumlahan *barcode A* yang di deteksi. Untuk lingkaran berwarna merah merupakan hasil jumlah barang yang terdeteksi oleh sensor. Selain itu, lingkaran berwarna hitam jumlah keseluruhan hasil deteksi barang pada *scanner*

barcode seperti Gambar 4.34. Hasil penjumlahan tersebut di dapatkan dari pengujian deteksi *barcode* A dengan 10 kali percobaan yang dilakukan secara berurutan untuk lebih jelasnya dapat dilihat pada Tabel 4.6.

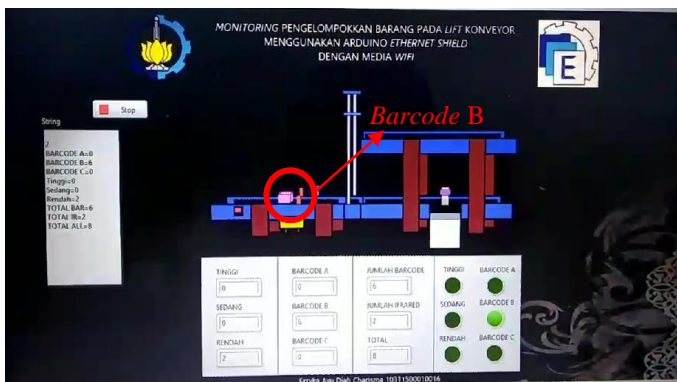
Tabel 4.6 Hasil Pengujian *Barcode A*

Barang <i>Barcode A</i> Ke-	Hasil Tampilan <i>Monitoring</i>	Keterangan
1	Barang Rendah	Tidak Sesuai
2	<i>Barcode A</i>	Sesuai
3	<i>Barcode A</i>	Sesuai
4	<i>Barcode A</i>	Sesuai
5	Barang Sedang	Tidak Sesuai
6	<i>Barcode A</i>	Sesuai
7	Barang Rendah	Tidak Sesuai
8	Barang Rendah	Tidak Sesuai
9	Barang Rendah	Tidak Sesuai
10	<i>Barcode A</i>	Sesuai

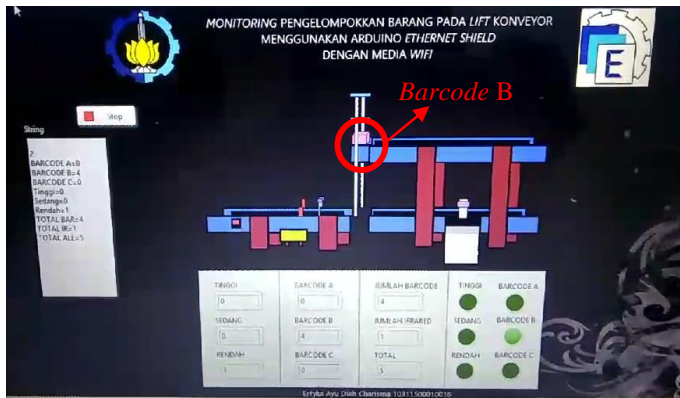
Berdasarkan Tabel 4.6 didapatkan bahwa, ketika *barcode A* terdeteksi di konveyor 1 maka kondisi ketika *relay 2* mengaktifkan motor pada konveyor 2 dan *relay 3* mengaktifkan motor pada konveyor 3 yang kemudian barang akan berjalan lurus. Dari Tabel 4.6 dapat dilihat ketika melakukan 10 kali percobaan terdapat kesalahan sebanyak 5 kali dan mendapatkan presentase *error* sebesar 50%. Hasil presentase *error* di dapatkan dari Persamaan 4.1. Untuk Tabel 4.6 pengujian dikatakan tidak sesuai apabila hasil *monitoring* pada LabVIEW mendeteksi *barcode A* Sedangkan pengujian dikatakan sesuai jika hasil *monitoring* pada LabVIEW mendeteksi *barcode A*.

Dari 10 percobaan tersebut, hanya 5 percobaan yang berhasil mendeteksi dengan benar karena *barcode* tidak terdeteksi oleh *scanner barcode* sehingga ketika melewati sensor *infrared* barang terdeteksi untuk dikelompokkan sesuai ketinggian. Sensor *infrared* dapat mendeteksi barang berlabel *barcode* walaupun barang berlabel *barcode* berwarna ungu, karena sensor *infrared* dapat mendeteksi barang dengan indikator warna hitam atau putih. Karena warna barang untuk barang berlabel *barcode* berwarna ungu muda dan dikarenakan pencahayaan yang terlalu terang atau gelap, maka sensor *infrared* mudah untuk mendeteksi barang berlabel *barcode* yang berwarna ungu tersebut.

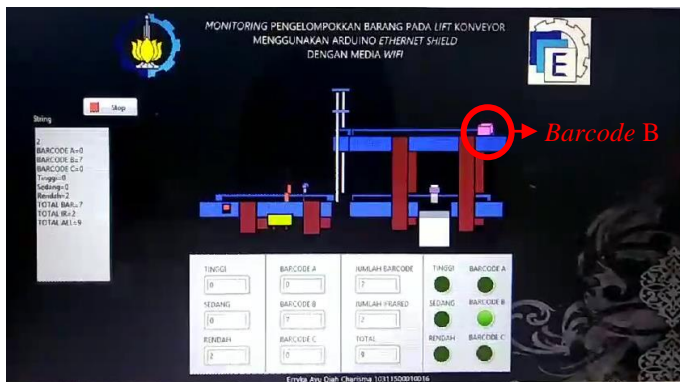
Setelah pengujian *barcode* A selesai, maka tekan tombol *reset* pada Arduino. Selanjutnya, *input* barang berlabel *barcode* B di konveyor maka barang akan berjalan mendekati *scanner barcode* hingga berbunyi “tit” yang menandakan *barcode* terdeteksi kemudian mengelompokkannya sesuai yang telah ditentukan. Ketika *barcode* B berjalan di konveyor 1 yang akan menuju ke konveyor 4 terlebih dahulu harus melewati konveyor 2 yang digunakan sebagai *lift* guna mengelompokkan barang untuk lebih jelasnya dapat dilihat pada Lampiran A.11 halaman A-14. Begitu pun untuk tampilan di LabVIEW, *barcode* A berjalan di konveyor 1 untuk menuju ke konveyor 4 terlebih dahulu harus melewati konveyor 2 yang digunakan sebagai *lift* seperti Gambar 3.35, Gambar 3.36. dan Gambar 3.37. Setelah proses pengelompokan selesai, maka LabVIEW akan menampilkan hasil perhitungan pada Gambar 3.38. Kemudian barang dapat di *input* kan kembali setelah proses pengelompokan selesai. Setelah *barcode* yang sama di *input* kan sebanyak 10 kali untuk pengujian, maka tekan tombol *reset* pada Arduino dan ambil data pengujian untuk *barcode* lainnya sebanyak 10 kali. Dan berikut ini adalah gambar *step-step* dari pengelompokan barang *barcode* B pada LabVIEW.



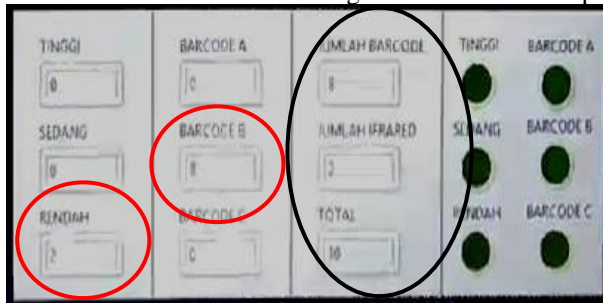
Gambar 4.35 Barang *Barcode* B Bergerak di Konveyor 1



Gambar 4.36 Barang Barcode B Naik Lift Melalui Konveyor 2



Gambar 4.37 Kondisi Ketika Barang Barcode B Dikelompokkan



Gambar 4.38 Hasil Penjumlahan Barcode B

Ketika *scanner barcode* di konveyor mendeteksi *barcode B* maka barang tersebut akan berjalan dari konveyor 1 hingga konveyor 4 melalui konveyor 2 yang sebagai *lift* proses tersebut dapat dilihat pada Lampiran A.11 halaman A-14 . Begitu pula pada tampilan LabVIEW, untuk Gambar 4.35, barang berjalan di konveyor 1 dan LED menyala di kondisi *barcode B*. Kemudian barang naik *lift* melalui konveyor 2 seperti pada Gambar 4.36. Disaat konveyor 2 sejajar dengan konveyor 4 maka barang berjalan ke konveyor 4 hingga sampai di ujung konveyor yang ditunjukkan pada Gambar 4.37. Setelah pengelompokkan *barcode B* pada LabVIEW telah selesai maka LED yang awalnya menyala di kondisi *barcode B* akan mati dan menampilkan hasil penjumlahan *barcode B* yang di deteksi. Untuk lingkaran berwarna merah merupakan hasil jumlah barang yang terdeteksi oleh sensor. Selain itu, lingkaran berwarna hitam jumlah keseluruhan hasil deteksi barang pada *scanner barcode* seperti Gambar 4.38. Hasil penjumlahan tersebut di dapatkan dari pengujian deteksi *barcode B* dengan 10 kali percobaan yang dilakukan secara berurutan untuk lebih jelasnya dapat dilihat pada Tabel 4.7.

Tabel 4.7 Hasil Pengujian *Barcode B*

Barang <i>Barcode B</i> Ke-	Hasil Tampilan <i>Monitoring</i>	Keterangan
1	<i>Barcode B</i>	Sesuai
2	<i>Barcode B</i>	Sesuai
3	<i>Barcode B</i>	Sesuai
4	<i>Barcode B</i>	Sesuai
5	Barang Rendah	Tidak Sesuai
6	<i>Barcode B</i>	Sesuai
7	<i>Barcode B</i>	Sesuai
8	Barang Rendah	Tidak Sesuai
9	<i>Barcode B</i>	Sesuai
10	<i>Barcode B</i>	Sesuai

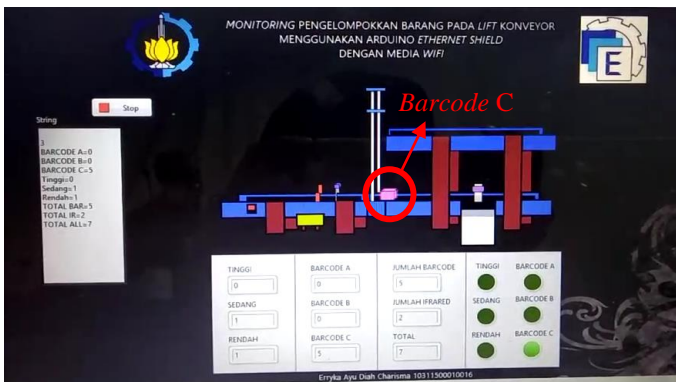
Berdasarkan Tabel 4.7 didapatkan bahwa, ketika *barcode B* terdeteksi di konveyor 1 maka kondisi ketika *relay 2* mengaktifkan motor konveyor 2 kemudian *relay 5* aktif untuk mengaktifkan motor pada *lift* agar dapat berjalan keatas atau naik, lalu *relay 2* mengaktifkan motor konveyor 2 dan *relay 4* untuk mengaktifkan motor konveyor 4. Ketika barang sudah dikelompokkan, secara otomatis *relay 6* aktif untuk mengaktifkan motor yang menurunkan *lift* agar kembali ke semula.

Dari Tabel 4.7 dapat dilihat ketika melakukan 10 kali percobaan terdapat kesalahan sebanyak 2 kali dengan presentase *error* sebesar 80%. Hasil presentase *error* di dapatkan dari Persamaan 4.1. Untuk Tabel 4.7 pengujian dikatakan tidak sesuai apabila hasil *monitoring* pada LabVIEW mendeteksi *barcode* B Sedangkan pengujian dikatakan sesuai jika hasil *monitoring* pada LabVIEW mendeteksi *barcode* B. Dari 10 percobaan tersebut, hanya 8 percobaan yang berhasil mendeteksi dengan benar karena *barcode* tidak terdeteksi oleh *scanner barcode* sehingga ketika melewati sensor *infrared* barang terdeteksi untuk dikelompokkan sesuai ketinggian. Sensor *infrared* dapat mendeteksi barang berlabel *barcode* walaupun barang berlabel *barcode* berwarna ungu, karena sensor *infrared* dapat mendeteksi barang dengan indikator warna hitam atau putih. Karena warna barang untuk barang berlabel *barcode* berwarna ungu muda dan dikarenakan pencahayaan yang terlalu terang atau gelap, maka sensor *infrared* mudah untuk mendeteksi barang berlabel *barcode* yang berwarna ungu tersebut.

Setelah pengujian *barcode* B selesai, maka tekan tombol *reset* pada Arduino. Selanjutnya, *input* barang berlabel *barcode* C di konveyor maka barang akan berjalan mendekati *scanner barcode* hingga berbunyi “tit” yang menandakan *barcode* terdeteksi kemudian mengelompokkannya sesuai yang telah ditentukan. Ketika *barcode* C berjalan di konveyor 1 dan menuju ke *pusher* yang terletak pada tengah-tengah konveyor 3 untuk lebih jelasnya dapat dilihat pada Lampiran A.12 halaman A-16. Begitu pun untuk tampilan di LabVIEW, *barcode* C berjalan di konveyor 1 dan menuju ke *pusher* seperti Gambar 4.39, Gambar 4.40, Gambar 4.41, dan Gambar 4.42. Setelah proses pengelompokan selesai, maka LabVIEW akan menampilkan hasil perhitungan pada Gambar 4.43. Kemudian, barang dapat di *input* kan kembali setelah proses pengelompokan selesai. Setelah *barcode* yang sama di *input* kan sebanyak 10 kali untuk pengujian dan ambil data pengujian untuk *barcode* C sebanyak 10 kali. Dan berikut ini adalah gambar *step-step* dari pengelompokan barang *barcode* C pada LabVIEW.



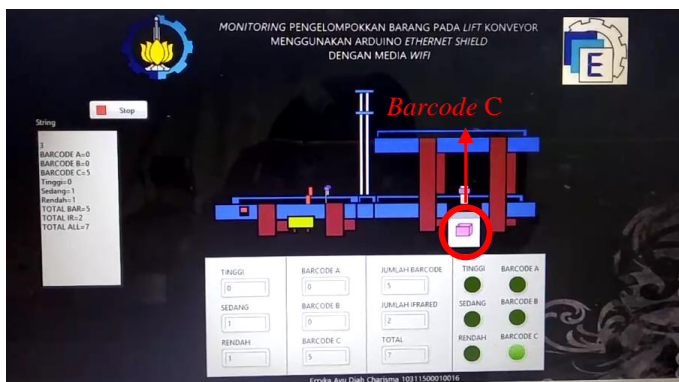
Gambar 4.39 Barang *Barcode C* Bergerak di Konveyor 1



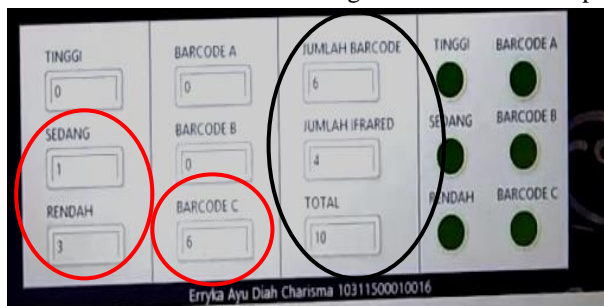
Gambar 4.40 Barang *Barcode C* Melewati Konveyor 2



Gambar 4.41 Pusher Mendorong Barang Barcode C



Gambar 4.42 Kondisi Ketika Barang Barcode C Dikelompokkan



Gambar 4.43 Hasil Penjumlahan Barcode C

Ketika *scanner barcode* di konveyor mendeteksi *barcode C* maka barang tersebut akan berjalan dari konveyor 1 ke *pusher*, proses tersebut dapat dilihat pada Lampiran A.12 halaman A-15. Begitu pula pada tampilan LabVIEW, untuk Gambar 4.39 barang berjalan di konveyor 1 dan LED menyala di kondisi *barcode C*. Kemudian barang berjalan melewati konveyor 2 seperti pada Gambar 4.40. Barang masih berjalan hingga menuju *pusher* dan barang tersebut di dorong oleh *pusher* yang ditunjukkan pada Gambar 4.41. Setelah pengelompokkan *barcode C* pada LabVIEW telah selesai maka LED yang awalnya menyala di kondisi *barcode C* akan mati dan menampilkan hasil penjumlahan *barcode C* yang di deteksi. Untuk lingkaran berwarna merah merupakan hasil jumlah barang yang terdeteksi oleh sensor. Selain itu, lingkaran berwarna hitam jumlah keseluruhan hasil deteksi barang pada *scanner barcode* seperti Gambar 4.43. Hasil penjumlahan tersebut di dapatkan dari pengujian deteksi *barcode C* dengan 10 kali percobaan yang dilakukan secara berurutan untuk lebih jelasnya dapat dilihat pada Tabel 4.8.

Tabel 4.8 Hasil Pengujian *Barcode C*

Barang <i>Barcode C</i> Ke-	Hasil Tampilan <i>Monitoring</i>	Keterangan
1	<i>Barcode C</i>	Sesuai
2	<i>Barcode C</i>	Sesuai
3	<i>Barcode C</i>	Sesuai
4	<i>Barcode C</i>	Sesuai
5	<i>Barcode C</i>	Sesuai
6	Barang Sedang	Tidak Sesuai
7	Barang Rendah	Tidak Sesuai
8	<i>Barcode C</i>	Sesuai
9	Barang Rendah	Tidak Sesuai
10	Barang Rendah	Tidak Sesuai

Berdasarkan Tabel 4.8 didapatkan bahwa, ketika *barcode B* terdeteksi di konveyor 1 maka kondisi *relay 2* mengaktifkan konveyor 2 dan *relay 3* aktif untuk mengaktifkan konveyor 3 kemudian *relay 7* aktif ketika barang telah sampai di depan *pusher*, kemudian *pusher* akan mendorongnya. Setelah selesai, *relay 8* akan aktif untuk mengembalikan *pusher* dalam keadaan semula.

Dari Tabel 4.8 dapat dilihat ketika melakukan 10 kali percobaan terdapat kesalahan sebanyak 4 kali dan mendapatkan

presentase *error* sebesar 60% yang didapatkan dari Persamaan 4.1. Untuk Tabel 4.7 pengujian dikatakan tidak sesuai apabila hasil *monitoring* pada LabVIEW mendeteksi *barcode* A. Berbeda dengan pengujian jika dikatakan sesuai apabila hasil *monitoring* pada LabVIEW mendeteksi *barcode* B. Dari 10 percobaan tersebut, hanya 6 percobaan yang berhasil mendeteksi dengan benar karena 2 percobaan pada *barcode* C tidak terdeteksi oleh *scanner barcode*. Sehingga ketika melewati sensor *infrared* barang terdeteksi untuk dikelompokkan sesuai ketinggian. Sensor *infrared* dapat mendeteksi barang berlabel *barcode* walaupun barang berlabel *barcode* berwarna ungu, karena sensor *infrared* dapat mendeteksi barang dengan indikator warna hitam atau putih. Karena warna barang untuk barang berlabel *barcode* berwarna ungu muda dan dikarenakan pencahayaan yang terlalu terang atau gelap, maka sensor *infrared* mudah untuk mendeteksi barang berlabel *barcode* yang berwarna ungu tersebut.

Dapat disimpulkan total pengujian dilakukan sebanyak 30 kali dan terdapat kesalahan sebanyak 11 kali. Untuk presentase *error* diperoleh sebesar 36% yang dikutip pada Persamaan 4.1. Kesalahan tersebut berupa salah mendeteksi jenis barang yang seharusnya dikelompokkan sesuai *barcode* menjadi berdasarkan ketinggian.

-----Halaman ini sengaja dikosongkan-----

BAB V

PENUTUP

Setelah dilakukan proses perencanaan, pembuatan dan pengujian alat serta membandingkan dengan teori-teori penunjang, dan dari data yang didapat sehingga mendapatkan kesimpulan dan saran dari kegiatan yang telah dilakukan.

5.1 Kesimpulan

Dari tugas akhir yang dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut :

1. Dalam pengujian terdapat kesalahan berupa ketidaksesuaian barang yang akan dikelompokkan pada konveyor dengan hasil *monitoring* pada LabVIEW. Ketidaksesuaian dalam hal barang sesuai ketinggian dan *barcode* yang di deteksi dengan hasil *monitoring* di LabVIEW.
2. Pengelompokkan barang dapat di *monitoring* dengan jarak hingga 120 meter tanpa adanya halangan. Jika dengan halangan dapat di *monitoring* dengan jarak hingga 15 meter.

Proyek TA ini merupakan lanjutan dari Tugas Akhir tahun 2010 dan mempunyai perbedaan dengan TA sebelumnya. Dimana terdapat perbedaan pada penggunaan *monitoring*. Pada proyek TA kali ini menggunakan *ethernet shield* dengan media *wifi* dan *software* LabVIEW.

5.2 Saran

Untuk menyempurnakan hasilnya perlu dipertimbangkan hal – hal sebagai berikut :

1. Perlu ditingkatkan dalam pembuatan menu-menu yang menarik pada tampilan *monitoring* di LabVIEW.
2. Untuk mengantisipasi komunikasi pada *monitoring* tidak terhambat dapat diterapkan saat melakukan *setting* komunikasi antara Arduino dan *ethernet shield* diperhatikan COM dan Alamat *IP* yang digunakan sesuai agar komunikasi dapat berjalan dengan lancar sehingga dapat terhubung dengan *wifi*. Kemudian dalam melakukan pemasangan kabel RJ45 harus lebih teliti karena jika terjadi kesalahan dalam penyambungan maka data tidak akan dapat terkirim sesuai dengan alamat yang di buat.

-----Halaman ini sengaja dikosongkan-----

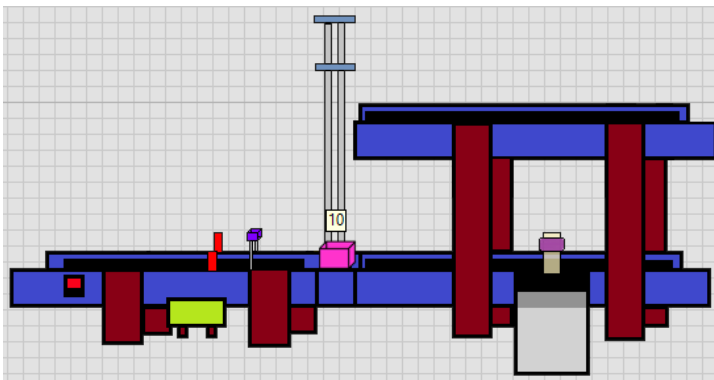
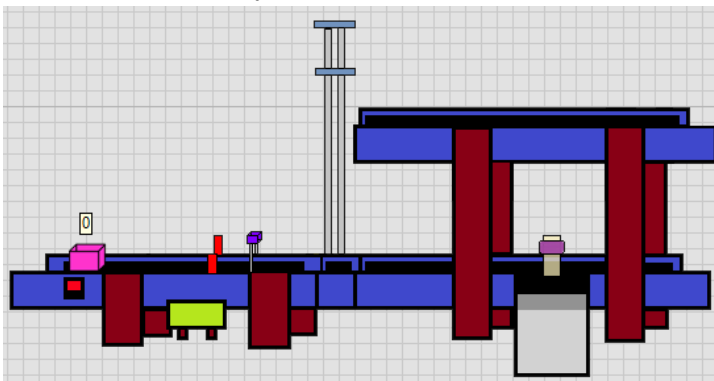
DAFTAR PUSTAKA

- [1] Deden Saputra, Monitoring dan Pengontralan Secara Terpusat Bak Penampung Air Menggunakan Flow Programming Berbasis Jaringan LAN. **Tugas Akhir**, Program Studi D4 Teknik Elektronika Jurusan Teknik Elektro-Politeknik Negeri Padang, Oktober 2017.
- [2] Nanda Lintang Syafitri, Sistem Komunikasi dan Monitoring Adaptif Relai Arus Lebih Terhadap Sumber Tegangan Pada Saluran Listrik Satu Fasa. **Tugas Akhir**, Departemen Teknik Elektro Otomasi FV-ITS, Juli 2017.
- [3] Hibatul, Wildan, Perencanaan Ulang Belt Conveyor A2 Pada PLTU PT. PJB PAITON Dengan Kapasitas 3500 Ton/Jam. **Tugas Akhir**, Program Studi D3 Teknik Mesin FTI-ITS, Juli 2013.
- [4] Fadila Lingga Dewi, Alat Penghitung Jumlah Barang Menggunakan Barcode ITF-14, **Tugas Akhir**, Departemen Teknik Elektro Otomasi FV-ITS, Juli 2017.
- [5] Andik Kurnia Adi Pratama dan Rahmat Bagus Prasajo, Mesin Pengelompok Barang Berdasarkan Ketinggian Dan Konveyor Pengangkut Menggunakan PLC, **Tugas Akhir**, Program Studi D3 Teknik Elektro FTI-ITS, Januari 2015.

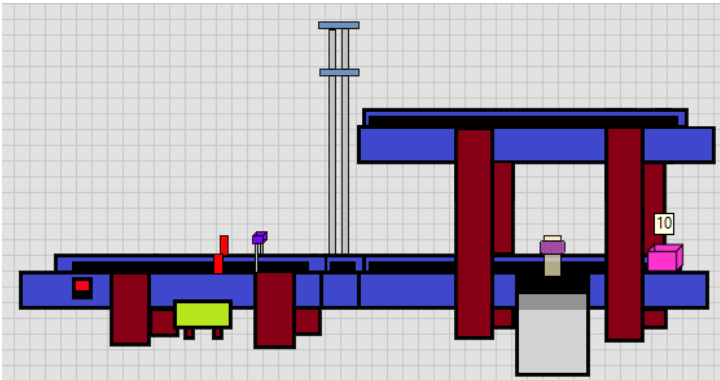
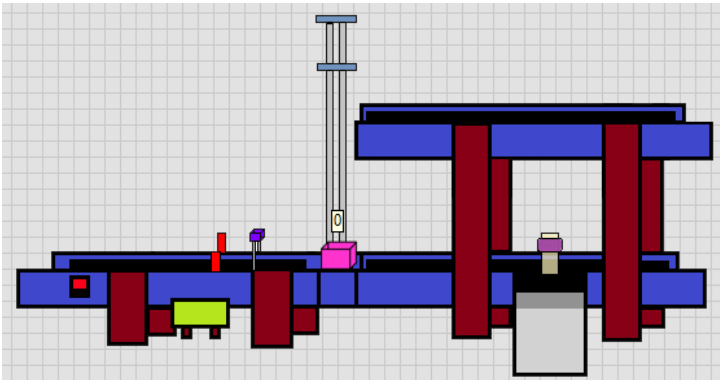
-----Halaman ini sengaja dikosongkan-----

LAMPIRAN A

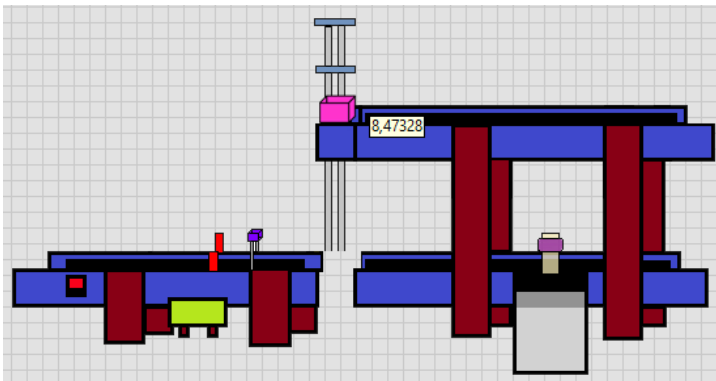
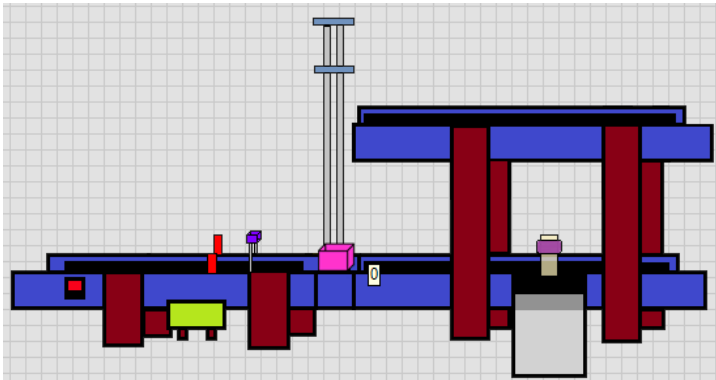
A.1 Foto Animasi Konveyor *ConvStart*



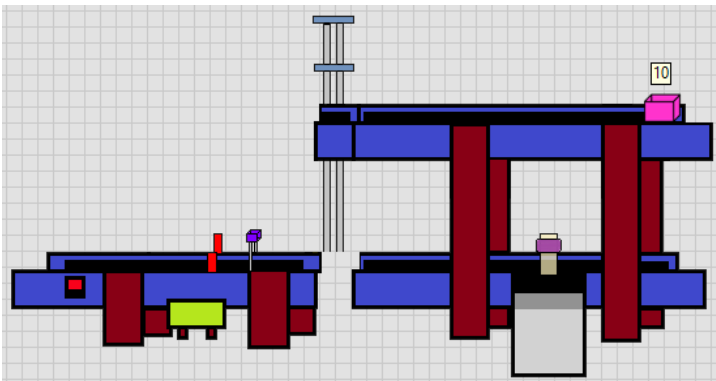
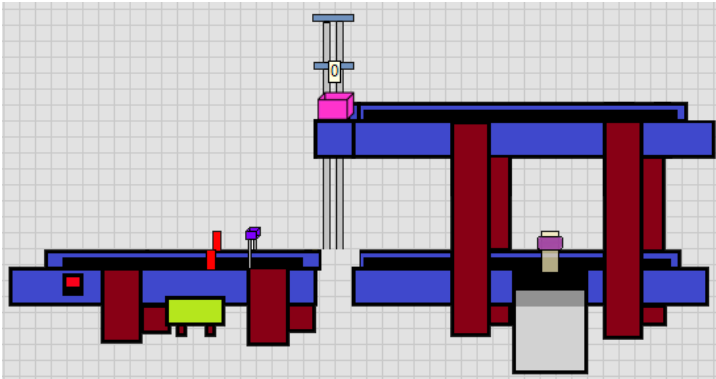
A.2 Foto Animasi Konveyor *ConvDown*



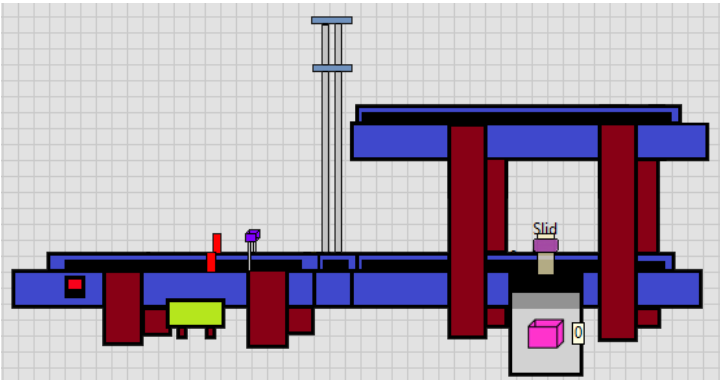
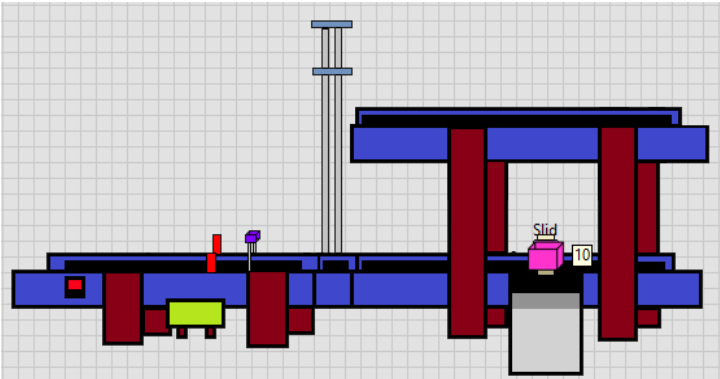
A.3 Foto Animasi Konveyor *ConvLift*



A.4 Foto Animasi Konveyor *ConvUp*

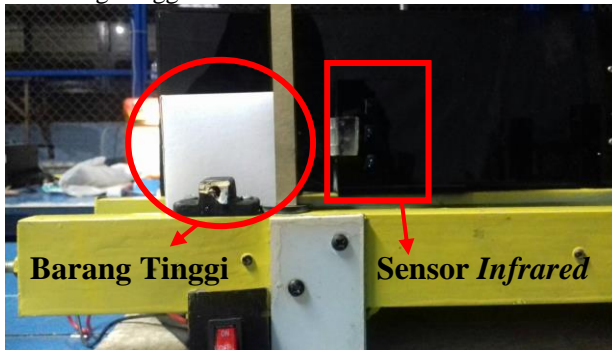


A.5 Foto Animasi Konveyor *ConvTrash*

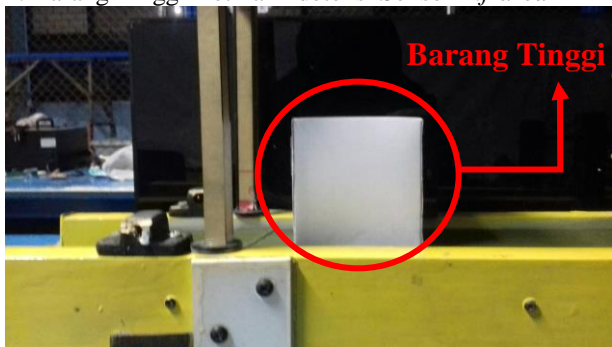


A.6 Step-Step Pengelompokkan Barang Tinggi pada Konveyor

1. Barang Tinggi Sebelum Dideteksi



2. Barang Tinggi Ketika Dideteksi Sensor Infrared

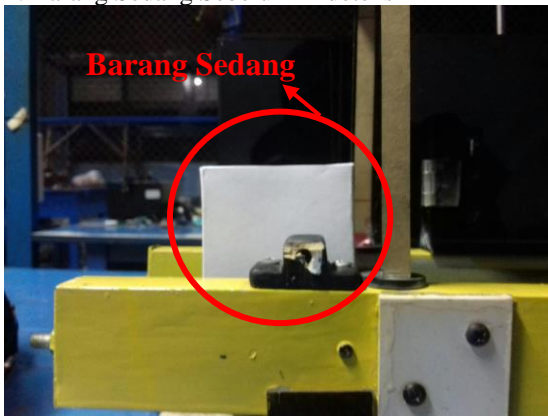


3. Kondisi Barang Tinggi Berada pada Ujung Konveyor 3 yang Dikelompokkan

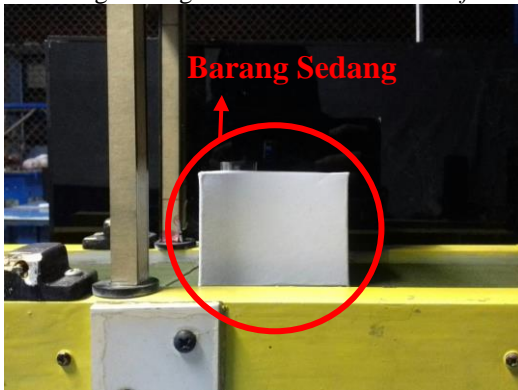


A.7 Step-Step Pengelompokkan Barang Sedang pada Konveyor

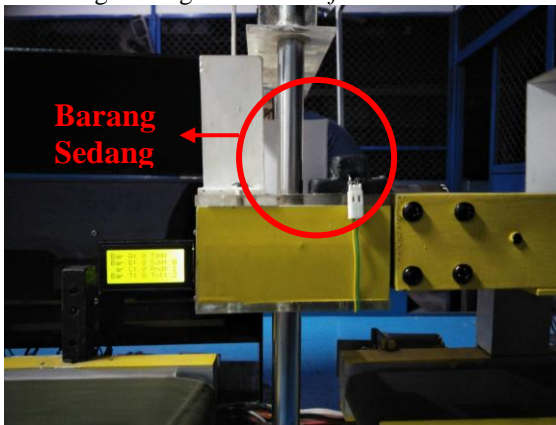
1. Barang Sedang Sebelum Dideteksi



2. Barang Sedang Dideteksi oleh Sensor *Infrared*



3. Barang Sedang Berada di *Lift*

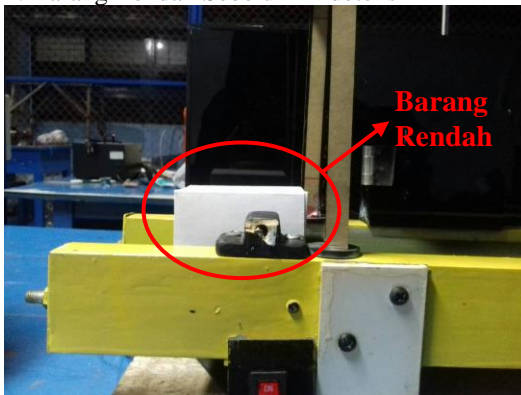


4. Kondisi Barang Sedang Berada di Ujung Konveyor 4 yang Dikelompokkan

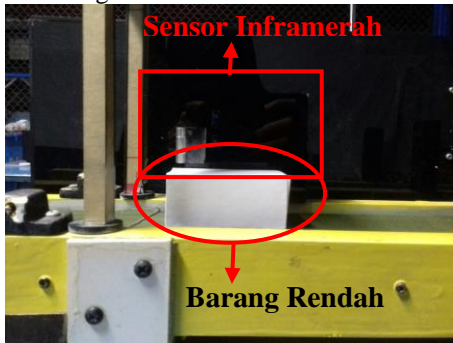


A.8 Step-Step Pengelompokkan Barang Rendah pada Konveyor

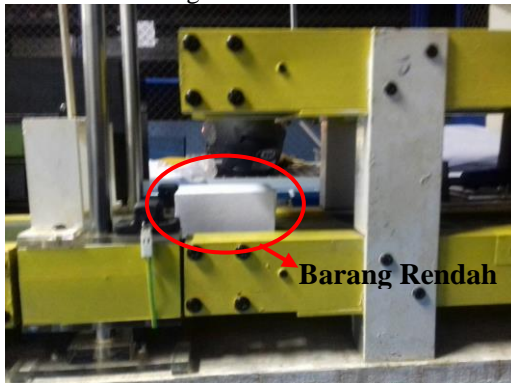
1. Barang Rendah Sebelum Dideteksi



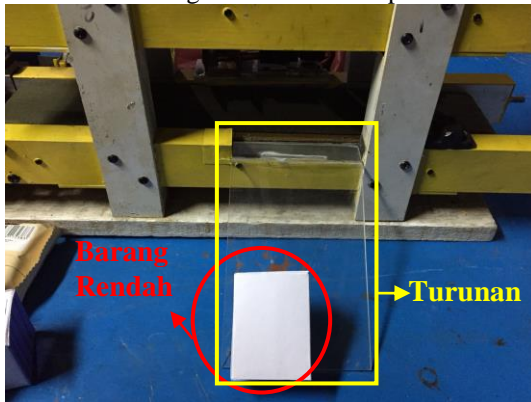
2. Barang Rendah Ketika Dideteksi



3. Kondisi Barang Rendah Ketika Didistribusikan



4. Kondisi Barang Rendah Dikelompokkan



A.9 Contoh Barang Berlabel *Barcode*



Contoh Barang *Barcode C*



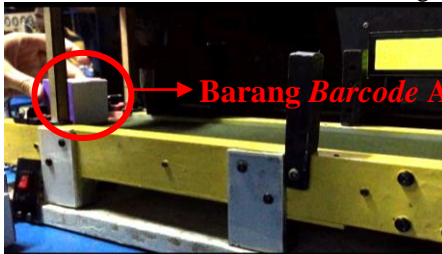
Contoh Barang *Barcode B*



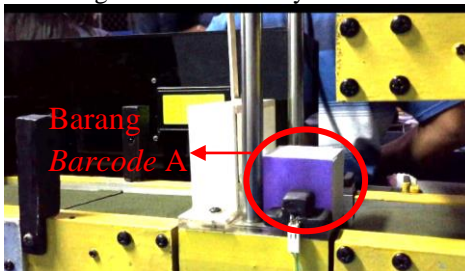
Contoh Barang *Barcode A*

A.10 Step-Step Pengelompokkan Barcode A pada Konveyor

1. Scanner Barcode Mendeteksi Barang Barcode A



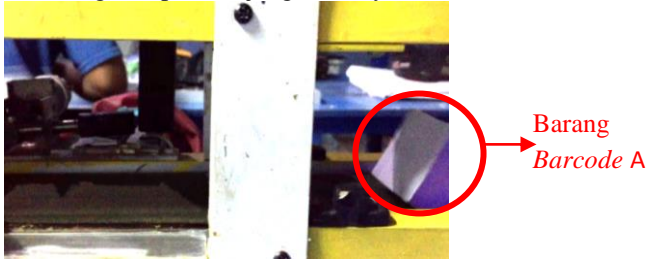
2. Barang Melewati Konveyor 2



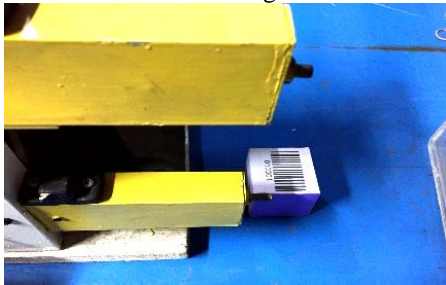
3. Barang Melewati Konveyor 3



4. Barang Sampai di Ujung Konveyor 3

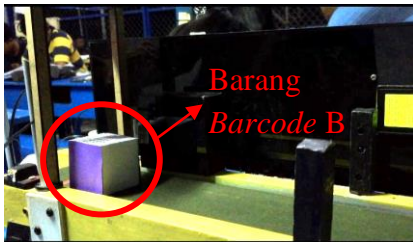


5. Kondisi Ketika Barang *Barcode A* Dikelompokkan

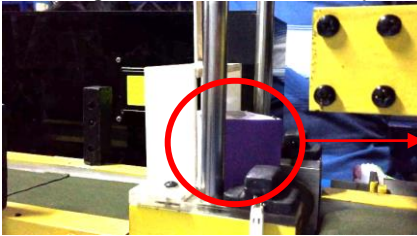


A.11 Step-Step Pengelompokkan *Barcode B* pada Konveyor

1. Scanner Barcode Mendeteksi *Barcode B*



2. Barang *Barcode B* Berada di *Lift* Melalui Konveyor 2



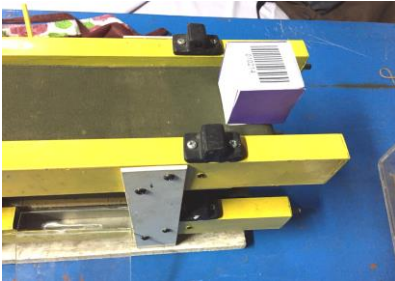
Barang *Barcode B*

3. Barang Melewati Konveyor 4



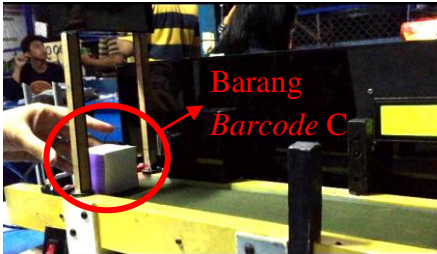
Barang *Barcode B*

4. Kondisi Ketika *Barcode B* Dikelompokkan



A.12 Step-Step Pengelompokkan *Barcode C* pada Konveyor

1. *Scanner Barcode* Mendeteksi *Barcode C*



2. *Pusher* Mendorong Barang *Barcode C*



3. Kondisi Ketika *Barcode C* Dikelompokkan



LAMPIRAN B

B.1 Pemrograman Proses Pengiriman Data Ke *Ethernet*

```
#include <Wire.h>
#include <LiquidCrystal.h>
#include <Ethernet.h>
#include <stdio.h>

LiquidCrystal lcd(23, 22, 17, 16, 15, 14);

byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
IPAddress ip(192, 168, 0, 175);
EthernetServer server(80);

#define relayon 0
#define relayoff 1

//String inputStringA = "00012345600012";      // a string to hold
incoming data
//String inputStringB = "12345678901217";
//String inputStringC = "30712345000010";

char stringA[8] = {'0', '1', '0', '3', '0', '1', '\r', 0};
char stringB[8] = {'0', '1', '0', '2', '1', '4', '\r', 0};
char stringC[8] = {'0', '1', '0', '3', '1', '5', '\r', 0};

char  inputString[14];

char lcd_buff[20]; //buffer karakter lcd

boolean stringCompleteA = false; // whether the string is complete
boolean stringCompleteB = false;
boolean stringCompleteC = false;
String barcode1, barcode2, barcode3;
int barcodeA = 0;
int barcodeB = 0;
int barcodeC = 0;
```

```

/*
int relay1 = 5;
int relay2 = 6;
int relay3 = 7;
int relay4 = 8;
int relay5 = 9;
int relay6 = 10;
int relay7 = 11;
int relay8 = 12;
*/

int relay1 = 28;
int relay2 = 30;
int relay3 = 32;
int relay4 = 34;
int relay5 = 36;
int relay6 = 38;
int relay7 = 40;
int relay8 = 42;

int h = 0, g = 0;
int n = 0;
char c;
char get_data[20];

char do_proses=0;

//variable infrared
char flag_ir=0;
char flag_set=0;
int datarendah, datasedang, datatinggi;
int flag_doProses;

const int DsensorTinggi = 2;
const int DsensorSedang = 3;
const int DsensorRendah = 8;
const int AsensorTinggi = A0;
const int AsensorSedang = A1;
const int AsensorRendah = A2;

```

```

int valueATinggi;
bool valueDTinggi;
int valueASedang;
bool valueDSedang;
int valueAREndah;
bool valueDREndah;

void setup()
{
  Serial.begin(115200);
  Serial1.begin(9600);
  Serial.println("INIT1");

  Ethernet.begin(mac, ip);
  server.begin();

  Serial.println("INIT2");
  //inputStringA.reserve(200);
  //inputStringB.reserve(200);
  //inputStringC.reserve(200);
  pinMode(relay1, OUTPUT);
  pinMode(relay2, OUTPUT);
  pinMode(relay3, OUTPUT);
  pinMode(relay4, OUTPUT);
  pinMode(relay5, OUTPUT);
  pinMode(relay6, OUTPUT);
  pinMode(relay7, OUTPUT);
  pinMode(relay8, OUTPUT);

  digitalWrite(relay1, relayoff);
  digitalWrite(relay2, relayoff);
  digitalWrite(relay3, relayoff);
  digitalWrite(relay4, relayoff);
  digitalWrite(relay5, relayoff);
  digitalWrite(relay6, relayoff);
  digitalWrite(relay7, relayoff);
  digitalWrite(relay8, relayoff);

```

```

lcd.begin(16, 4);
lcd.setCursor(4,0);
lcd.print("MONITORING");
lcd.setCursor(4,1);
lcd.print("BARANG DI");
lcd.setCursor(5,2);
lcd.print("KONVEYOR");
delay(5000);

}

void serialEvent1() //get barcode
{
  char inA = 1;
  char inB = 1;
  char inC = 1;

  if (Serial1.available())
  { // If anything comes in Serial1 (pins 0 & 1)
    //Serial.write(Serial1.read());
    c = (char)Serial1.read();

    if (c == '\n')
    {
      //Serial.println("printing");
      while (n >= 0)
      {
        if (inA == 1 && get_data[n] == stringA[n])
        {
          //Serial.write(stringA[n]);
          //Serial.println("cocok A");
        }
        else inA = 0;

        if (inB == 1 && get_data[n] == stringB[n])
        {

        } else inB = 0;
      }
    }
  }
}

```

```

    if (inC == 1 && get_data[n] == stringC[n])
    {

        } else inC = 0;
        Serial.write(get_data[n]);
        n--;
    }
    if (inA == 1)
    {
        stringCompleteA = true;
        Serial.println("\nbarcode A");
    }
    if (inB == 1)
    {
        stringCompleteB = true;
        Serial.println("\nbarcode B");
    }
    if (inC == 1)
    {
        stringCompleteC = true;
        Serial.println("\nbarcode C");
    }
    n = 0;
    //Serial1.flush();
    while (Serial1.available() > 0)
    {
        char k = Serial1.read();
        //w++;
        delay(1);
    }
    Serial1.end();delay(100);
    Serial1.begin(9600);

}
else
{
    get_data[n] = c;
    n++;
}

```

```

    }
}
void get_infrared()
{
    //int flag=0;valueATinggi = analogRead(AsensorTinggi);
    valueDTinggi = digitalRead(DsensorTinggi);
    valueASedang = analogRead(AsensorSedang);
    valueDSedang = digitalRead(DsensorSedang);
    valueARendah = analogRead(AsensorRendah);
    valueDRendah = digitalRead(DsensorRendah);

    if(valueDRendah==0 && valueDSedang==1 && valueDTinggi==1)
    {
        flag_ir=3; //ir rendah
        delay(10);
    }
    else if(valueDRendah==0 && valueDSedang==0 &&
valueDTinggi==1)
    {
        flag_ir=2; //ir sedang
        delay(10);
    }
    else if(valueDRendah==0 && valueDSedang==0 &&
valueDTinggi==0)
    {
        flag_ir=1; //ir tinggi
        delay(10);
    }
    else if(valueDRendah==1 && valueDSedang==1 &&
valueDTinggi==1)
    {
        flag_ir=0;
        delay(10);
    }
}
void proses_decision()
{
    get_infrared();
    Serial.print("  flag_ir >>");

```

```

Serial.println(flag_ir,DEC);

if((stringCompleteA) && flag_set==0)
{
    flag_set=1;
    do_proses=1;
}
else if((stringCompleteB) && flag_set==0)
{
    //client.print("2");
    flag_set=1;
    do_proses=2;
}
else if((stringCompleteC) && flag_set==0)
{
    //client.print("3");
    flag_set=1;
    do_proses=3;
}
else if(flag_ir==1 && flag_set==0)
{
    flag_set=1;
    do_proses=4;
}
else if(flag_ir==2 && flag_set==0)
{
    flag_set=1;
    do_proses=5;
}
else if(flag_ir==3 && flag_set==0)
{
    flag_set=1;
    do_proses=6;
}
else
{
    //do_proses=0;
    //flag_set=0; // comment this line when connected to ethernet client
}

```

```

flag_ir=0;

stringCompleteA=false;
stringCompleteB=false;
stringCompleteC=false;

Serial.print("proses decision> ");
Serial.println(do_proses,DEC);

delay(20);
}

void tampil_lcd()
{
    sprintf(lcd_buff,"Bar A:%2d Tgg:%2d",barcodeA,datatinggi); // untuk
mengkopi string ke var lcd_buff
    lcd.setCursor(0, 0);
    lcd.print(lcd_buff);

    sprintf(lcd_buff,"Bar B:%2d Sdg:%2d",barcodeB,datasedang);
    lcd.setCursor(0, 1);
    lcd.print(lcd_buff);

    sprintf(lcd_buff,"Bar C:%2d Rnd:%2d",barcodeC,datarendah);
    lcd.setCursor(0, 2);
    lcd.print(lcd_buff);

    sprintf(lcd_buff,"Bar T:%2d
Tot:%2d",barcodeC+barcodeB+barcodeA,datarendah+datasedang+datat
inggi);
    lcd.setCursor(0, 3);
    lcd.print(lcd_buff);

}

void loop()
{

```



```
///get_infrared();

proses_decision();
send2ethernet();
tampil_lcd();

if(flag_doProses>=100) //3dtk 60*50ms
{
    proses(do_proses);
    do_proses=0;
    flag_set=0;
    flag_doProses=0;
    Serial.print("proses ");
    Serial.print(do_proses);
    Serial.println("is done");
}
```

B.2 Pemrograman Proses Pada *Ethernet*

```
void send2ethernet()
{
  EthernetClient client = server.available();
  if (client)
  {
    flag_doProses=100;
    Serial.println("new client");
    // an http request ends with a blank line
    boolean currentLineIsBlank = true;
    while (client.connected())
    {
      if (client.available())
      {
        char c = client.read();
        Serial.write(c);

        if (c == '\n')
        {
          client.print("\n");
          //client_send();

          { //client send
            client.println(do_proses,DEC);

            Serial.println("send to client");
            client.print("BARCODE A=");
            client.println(barcodeA);

            client.print("BARCODE B=");
            client.println(barcodeB);

            client.print("BARCODE C=");
            client.println(barcodeC);

            client.print("Tinggi=");
            client.println(datatinggi);
```

```

        client.print("Sedang=");
        client.println(datasedang);

        client.print("Rendah=");
        client.println(datarendah);

        client.print("TOTAL BAR=");
        client.println(barcodeA + barcodeB + barcodeC);
        client.print("TOTAL IR=");
        client.println(datarendah+datasedang+datatinggi);
        client.print("TOTAL ALL=");
        client.println(datarendah+datasedang+datatinggi+barcodeA +
barcodeB + barcodeC);
    }
    break;
}
}
}
// give the web browser time to receive the data
delay(10);
client.stop();

Serial.print("do proses >>");
Serial.println(do_proses,DEC);

//proses(do_proses);
//do_proses=0;
//flag_set=0;
// close the connection:
Serial.println("client disconnected");
}

flag_doProses++;
delay(10);
}

```

B.3 Pemograman Untuk Pengelompokkan Pada Konveyor

```
void proses(int pr)
{
    if (pr==3||pr==6)
    {

        //Serial.println(inputStringA);
        //barcode1 = inputStringA.substring(0, 14);
        Serial.println("Deteksi Barcode A");
        Serial.print(barcode1);
        delay(1000);
        lcd.setCursor(14, 0);
        lcd.print(" ");
        //barcodeA++;
        Serial.print(barcode1);
        Serial.println(barcodeA);
        digitalWrite(relay2, relayon);
        delay(3000);
        digitalWrite(relay3, relayon);
        delay(500);
        digitalWrite(relay2, relayoff);
        delay(2000);
        digitalWrite(relay3, relayoff);
        delay(1000);
        digitalWrite(relay7, relayon);
        delay(2000);
        digitalWrite(relay7, relayoff);
        delay(1000);
        digitalWrite(relay8, relayon);
        delay(2000);
        digitalWrite(relay8, relayoff);
        delay(1000);
        //inputStringA = "00012345600012";
        stringCompleteA = false;
        Serial.println("A COMPLETED");

    }
}
```

```

else if (pr==2||pr==5)
{

    //Serial.println(inputStringB);
    //barcode2 = inputStringB.substring(0, 14);
    Serial.println("Deteksi Barcode B");
    Serial.print(barcode2);
    delay(1000);
    lcd.setCursor(14, 0);
    lcd.print(" ");
    //barcodeB++;
    Serial.print(barcode2);
    Serial.println(barcodeB);
    digitalWrite(relay2, relayon);
    delay(1900);
    digitalWrite(relay2, relayoff);
    delay(3000);
    digitalWrite(relay5, relayon);
    delay(14800);
    digitalWrite(relay5, relayoff);
    delay(1000);
    digitalWrite(relay2, relayon);
    delay(1000);
    digitalWrite(relay4, relayon);
    delay(1200);
    digitalWrite(relay2, relayoff);
    delay(3000);
    digitalWrite(relay4, relayoff);
    delay(1000);
    digitalWrite(relay6, relayon);
    delay(10900);
    digitalWrite(relay6, relayoff);
    delay(3000);
    //inputStringB = "12345678901217";
    stringCompleteB = false;
    Serial.println("B COMPLETED");
}

else if (pr==1||pr==4)

```

```

{
  //EthernetClient client = server.available();
  //client.print("PROSES C");

  //Serial.println(inputStringC);
  //barcode3 = inputStringC.substring(0, 14);
  Serial.println("Deteksi Barcode C");
  Serial.print(barcode3);
  delay(1000);
  lcd.setCursor(14, 0);
  lcd.print(" ");
  //barcodeC++;
  Serial.print(barcode3);
  Serial.println(barcodeC);
  digitalWrite(relay2, relayon);
  delay(2000);
  digitalWrite(relay3, relayon);
  delay(3500);
  digitalWrite(relay2, relayoff);
  delay(2000);
  digitalWrite(relay3, relayoff);
  delay(1000);
  //inputStringC = "30712345000010";
  stringCompleteC = false;
  Serial.println("C COMPLETED");
}

else
{
  digitalWrite(relay2, relayoff);
  digitalWrite(relay1, relayoff);
  digitalWrite(relay3, relayoff);
  digitalWrite(relay4, relayoff);
  digitalWrite(relay5, relayoff);
  digitalWrite(relay6, relayoff);
  digitalWrite(relay7, relayoff);
  digitalWrite(relay8, relayoff);
}

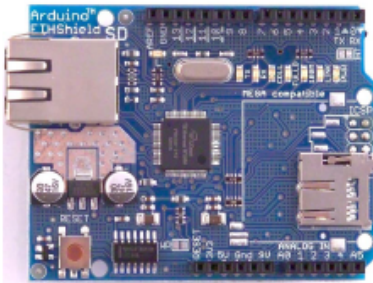
```

```
if(pr==1)barcodeA++;  
else if(pr==2)barcodeB++;  
else if(pr==3)barcodeC++;  
else if(pr==4)datatinggi++;  
else if(pr==5)datasedang++;  
else if(pr==6)datarendah++;  
  
flag_set=1;  
}
```


LAMPIRAN C

C.1 Datasheet Ethernet Shield

Arduino Ethernet Shield



Download: [arduino-ethernet-shield-05-schematic.pdf](#), [arduino-ethernet-shield-05-reference-design.zip](#)

Download: [arduino-ethernet-shield-schematic.pdf](#), [arduino-ethernet-shield-reference-design.zip](#)

The Arduino Ethernet Shield allows an Arduino board to connect to the internet. It is based on the [Wiznet W5100](#) ethernet chip ([datasheet](#)). The Wiznet W5100 provides a network (IP) stack capable of both TCP and UDP. It supports up to four simultaneous socket connections. Use the [Ethernet library](#) to write sketches which connect to the internet using the shield. The ethernet shield connects to an Arduino board using long wire-wrap headers which extend through the shield. This keeps the pin layout intact and allows another shield to be stacked on top.

The latest revision of the shield adds a micro-SD card slot, which can be used to store files for serving over the network. It is compatible with the Arduino Duemilanove and Mega (using the Ethernet library coming in Arduino 0019). An SD card library is not yet included in the standard Arduino distribution, but the [sdfatlib](#) by Bill Greiman works well. See [this tutorial from Adafruit Industries](#) for instructions (thanks Limor!).

The latest revision of the shield also includes a reset controller, to ensure that the W5100 Ethernet module is properly reset on power-up. Previous revisions of the shield were not compatible with the Mega and need to be manually reset after power-up. The original revision of the shield contained a full-size SD card slot; this is not supported.

Arduino communicates with both the W5100 and SD card using the SPI bus (through the ICSP header). This is on digital pins 11, 12, and 13 on the Duemilanove and pins 50, 51, and 52 on the Mega. On both boards, pin 10 is used to select the W5100 and pin 4 for the SD card. These pins cannot be used for general I/O. On the Mega, the hardware SS pin, 53, is not used to select either the W5100 or the SD card, but it must be kept as an output or the SPI interface won't work.

Note that because the W5100 and SD card share the SPI bus, only one can be active at a time. If you are using both peripherals in your program, this should be taken care of by the corresponding libraries. If you're not using one of the peripherals in your program, however, you'll need to explicitly deselect it. To do this with the SD card, set pin 4 as an output and write a high to it. For the W5100, set digital pin 10 as a high output.

The shield provides a standard RJ45 ethernet jack.

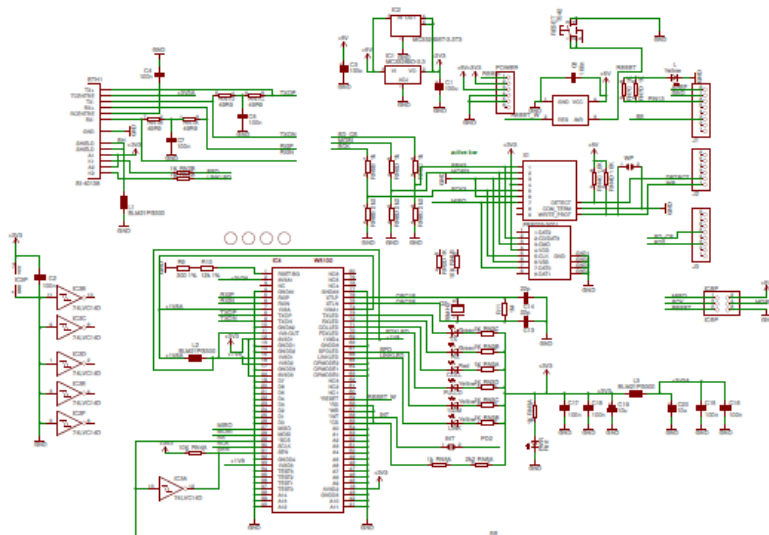
The reset button on the shield resets both the W5100 and the Arduino board.

The shield contains a number of informational LEDs:

- PWR: indicates that the board and shield are powered
- LINK: indicates the presence of a network link and flashes when the shield transmits or receives data
- FULLD: indicates that the network connection is full duplex
- 100M: indicates the presence of a 100 Mb/s network connection (as opposed to 10 Mb/s)
- RX: flashes when the shield receives data
- TX: flashes when the shield sends data
- COLL: flashes when network collisions are detected

The solder jumper marked "INT" can be connected to allow the Arduino board to receive interrupt-driven notification of events from the W5100, but this is not supported by the Ethernet library. The jumper connects the INT pin of the W5100 to digital pin 2 of the Arduino.

See also: [getting started with the ethernet shield](#) and [Ethernet library reference](#)



Arduino ETHERNET - shield V5

Copyright (c) 2010 Arduino
Released under the Creative Commons Attribution-Share Alike 3.0 License
<http://creativecommons.org/licenses/by-sa/3.0/>

C.2 Datasheet LM7808



Continental Device India Limited

An ISO/TS 16949, ISO 9001 and ISO 14001 Certified Company



3-TERMINAL POSITIVE VOLTAGE REGULATOR

LM7808

TO-220
Plastic Package



The Voltages available allow these Regulators to be used in Logic Systems, Instrumentation, HI-FI Audio Circuits and other Solid State Electronic Equipment

ABSOLUTE MAXIMUM RATINGS ($T_A=25^\circ\text{C}$)

DESCRIPTION	SYMBOL	VALUE	UNIT
Input Voltage	V_{IN}	35	V
Power Dissipation at $T_A=25^\circ\text{C}$	P_D	2	W
Power Dissipation at $T_J=25^\circ\text{C}$	P_D	15	W
Operating Free Air, Case, or Virtual Junction Temperature Range	T_J	0 to +150	$^\circ\text{C}$
Storage Temperature Range	T_{stg}	- 65 to +150	$^\circ\text{C}$
Lead Temperature 1.6mm (1/16 Inoh) from Case for 10 seconds	T_L	260	$^\circ\text{C}$

Recommended Operating Conditions

DESCRIPTION	SYMBOL	MIN	TYP	MAX	UNIT
Input Voltage	V_I	10.5		25	V
Output Current	I_O			1.5	A
Operating Junction Temperature	T_J	0		125	$^\circ\text{C}$

ELECTRICAL CHARACTERISTICS ($T_A=25^\circ\text{C}$ unless specified otherwise)

$V_{IN}=14\text{V}$, $I_O=500\text{mA}$, $T_A=25^\circ\text{C}$

DESCRIPTION	SYMBOL	TEST CONDITION	MIN	TYP	MAX	UNIT
Output Voltage	V_O	25°C $I_O=5\text{mA} \sim 1\text{A}$	7.7		8.3	V
		$V_{IN}=10.5 \sim 23\text{V}$, $P_D \leq 15\text{W}$, $0^\circ\text{C} \sim 125^\circ\text{C}$	7.6		8.4	V
Line Regulation	R_{REG}	$V_{IN}=10.5 \sim 25\text{V}$, 25°C			160	mV
		$V_{IN}=11 \sim 17\text{V}$, 25°C			80	mV
Ripple Rejection	R_R	$V_{IN}=11.5 \sim 21.5\text{V}$, $f=120\text{Hz}$, $0^\circ\text{C} \sim 125^\circ\text{C}$	55			dB
Load Regulation	R_{REG}	$I_O=5\text{mA} \sim 1.5\text{A}$, 25°C			160	mV
		$I_O=250\text{mA} \sim 750\text{mA}$, 25°C			80	mV
Output Resistance	r_O	$f=1\text{KHz}$, $0^\circ\text{C} \sim 125^\circ\text{C}$	0.016			Ω
Temperature Coefficient of Output Voltage	$\Delta V_O/\Delta T$	$I_O=5\text{mA}$, $0^\circ\text{C} \sim 25^\circ\text{C}$	-0.8			mV/ $^\circ\text{C}$
Output Noise Voltage	V_{NO}	$f=10\text{Hz} \sim 100\text{KHz}$, 25°C		52		μV
Dropout Voltage	$V_{DOF (max)}$	$I_O=1\text{A}$, 25°C		2.0		V
Quiescent Current	I_Q	25°C			8.0	mA
Quiescent Current Change	ΔI_{QH}	$V_{IN}=10.5 \sim 25\text{V}$, $0^\circ\text{C} \sim 150^\circ\text{C}$			1.0	mA
		$I_O=5\text{mA} \sim 1\text{A}$, $0^\circ\text{C} \sim 150^\circ\text{C}$			0.5	mA
Short Circuit Output Current	I_{SC}	25°C		450		mA
Peak Output Current	I_{Omax}	25°C		2.2		A

LM7808Rev_1 070306E

Component Disposal Instructions

1. CDIL Semiconductor Devices are RoHS compliant, customers are requested to please dispose as per prevailing Environmental Legislation of their Country.
2. In Europe, please dispose as per EU Directive 2002/96/EC on Waste Electrical and Electronic Equipment (WEEE).

Disclaimer

The product information and the selection guides facilitate selection of the CDIL's Semiconductor Device(s) best suited for application in your product(s) as per your requirement. It is recommended that you completely review our Data Sheet(s) so as to confirm that the Device(s) meet functionality parameters for your application. The information furnished in the Data Sheet and on the CDIL Web Site/CD are believed to be accurate and reliable. CDIL however, does not assume responsibility for inaccuracies or incomplete information. Furthermore, CDIL does not assume liability whatsoever, arising out of the application or use of any CDIL product; neither does it convey any license under its patent rights nor rights of others. These products are not designed for use in life saving/support appliances or systems. CDIL customers selling these products (either as individual Semiconductor Devices or incorporated in their end products), in any life saving/support appliances or systems or applications do so at their own risk and CDIL will not be responsible for any damages resulting from such sale(s).

CDIL strives for continuous improvement and reserves the right to change the specifications of its products without prior notice.



CDIL is a registered Trademark of
Continental Device India Limited

G-120 Naraina Industrial Area, New Delhi 110 028, India.

Telephone + 91-11-2579 6150, 4141 1112 Fax + 91-11-2579 5290, 4141 1119
email@cdil.com www.cdilsemi.com

C.3 Datasheet LM7908



TIGER ELECTRONIC CO.,LTD

Product specification

3-Terminal 1A Positive Voltage Regulator

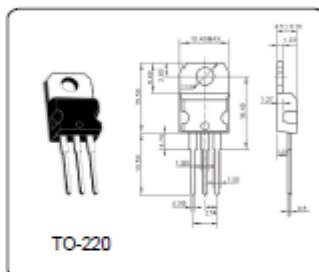
LM7908

GENERAL DESCRIPTION

The LM7908 series of three terminal positive regulators are available in the TO-220 package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to

ABSOLUTE MAXIMUM RATINGS ($T_a = 25^\circ\text{C}$)

Parameter	Symbol	Typ	Unit
Input Voltage	V_i	-35	V
Output Voltage	V_o	-8.0	V
Peak Current	I_{pk}	-2.2	A
Operating Temperature Range	T_{opr}	0~125	$^\circ\text{C}$
Storage Temperature Range	T_{sto}	-65~150	$^\circ\text{C}$



ELECTRICAL CHARACTERISTICS ($T_a = 25^\circ\text{C}$)

(Refer to test circuit, $I_o = -500\text{mA}$, $V_i = -14\text{V}$, $C_i = 2.2\mu\text{F}$, $C_o = 1.0\mu\text{F}$ unless otherwise specified)

Parameter	Symbol	Test Conditions	Min	Typ	Max	Unit
Output Voltage	V_o	$V_i = -10.5\text{V to } -21\text{V}$	-7.76	-8.0	-8.24	V
Line Regulation (Note1)	Regline	$V_i = -10.5\text{V to } -25\text{V}$	—	-10	-160	mV
		$V_i = -11\text{V to } -17\text{V}$	—	-5.0	-80	
Load Regulation (Note1)	Regload	$I_o = -5.0\text{mA to } -1.5\text{A}$	—	-12	-160	mV
		$I_o = -250\text{mA to } -750\text{mA}$	—	-4	-80	
Quiescent Current	I_Q	$T_j = +25^\circ\text{C}$	—	-3	6	mA
Quiescent Current Change	ΔI_Q	$I_o = -5\text{mA to } -1\text{A}$	—	—	-0.5	mA
		$V_i = -11.5\text{V to } -25\text{V}$	—	—	-1.0	
Ripple Rejection	RR	$f = 120\text{Hz}$, $V_o = -11\text{V to } -21\text{V}$	54	60	—	dB
Dropout Voltage	V_{drop}	$I_o = -1\text{A}$, $T_j = +25^\circ\text{C}$	—	-2	—	V
Short Circuit Current	I_{sc}	$V_i = -35\text{V}$, $T_a = +25^\circ\text{C}$	—	-300	—	mA
Peak Current	I_{pk}	$T_j = +25^\circ\text{C}$	—	-2.2	—	A

-----Halaman ini sengaja dikosongkan-----

DAFTAR RIWAYAT HIDUP



Nama : Erryka Ayu Diah
Charisma
TTL : Surabaya, 19
September 1996
Jenis Kelamin : Perempuan
Agama : Islam
Alamat Rumah : Manukan Rukun II
Blok 18C/16,
Surabaya
Telepon/HP : 081803334773
E-mail : erryka19@gmail.com

RIWAYAT PENDIDIKAN

1. 2000-2002
TK Kurnia
2. 2002-2008
SDN Manukan Kulon III/540,
Surabaya
SMPN 20 Surabaya
3. 2008-2011
SMAN 11 Surabaya
4. 2011-2014
Departemen Teknik Elektro
Otomasi Prodi Studi Elektro
Industri Fakultas Vokasi (FV)
5. 2015-2018
Institut Teknologi Sepuluh
Nopember (ITS)

PENGALAMAN KERJA

1. Kerja Praktek di Pusat Veteriner Farma Juli – Agustus 2016
2. Kerja Praktek di PT. Campina Ice Cream Industry Juni – Juli 2017

PENGALAMAN ORGANISASI

1. Staff Kesejahteraan Mahasiswa (KESMA) Himpunan Mahasiswa D3 Teknik Elektro ITS 2016-2017
2. Kabiro Akademik Kesejahteraan Mahasiswa (KESMA) Himpunan Mahasiswa D3 Teknik Elektro ITS 2017-2018

-----Halaman ini sengaja dikosongkan-----